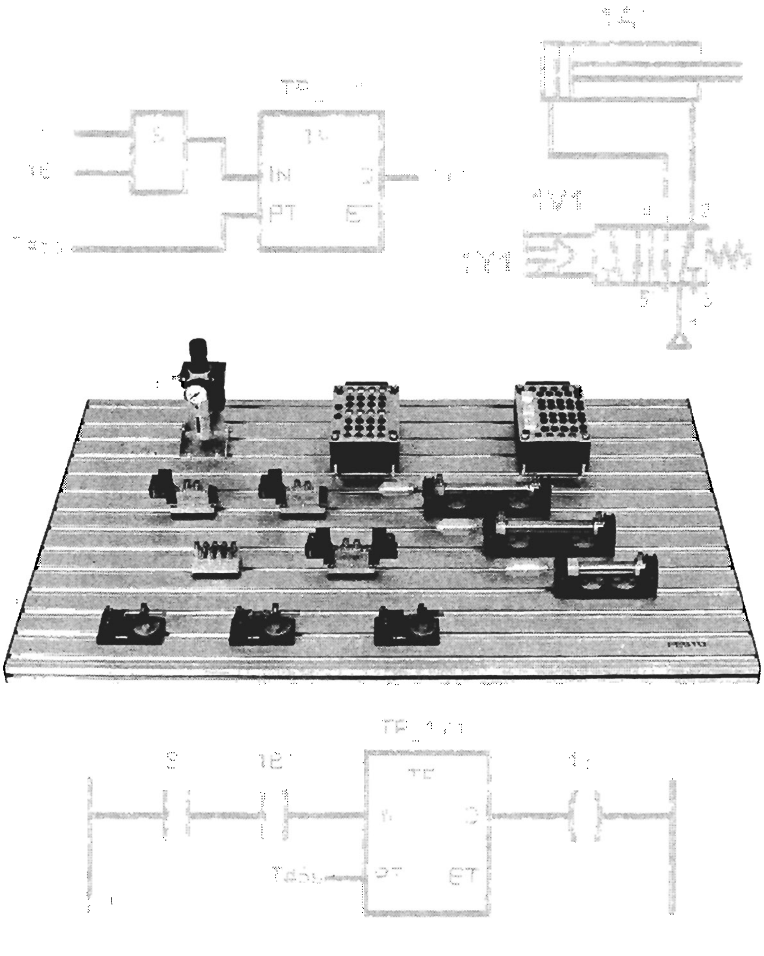
Learning System for Automation and Technology

Programmable logic controllers

Workbook Basic level



FESTO

093314

Authorised applications and liability

The Learning System for Automation and Technology has been developed and prepared exclusively for training in the field of automation. The training organization and / or trainee shall ensure that the safety precautions described in the accompanying Technical documentation are fully observed.

Festo Didactic hereby excludes any liability for injury to trainees, to the training organization and / or to third parties occurring as a result of the use or application of the station outside of a pure training situation, unless caused by premeditation or gross negligence on the part of Festo Didactic.

Order No.:	093314
Description:	ARBB.SPS FPC GS
Designation:	D.S301-C-FPC-A-GB
Edition:	08/2002
Layout:	28.08.2002 Beatrice Huber
Graphics:	Doris Schwarzenberger
Authors:	Ekhard v. Terzi, Christine Löffler, Frank Ebel

Preface

The Festo Didactic Learning System for Automation and Technology is designed to meet a number of different training and vocational requirements, and the training packages are structured accordingly:

- Basic packages convey basic knowledge spanning a wide range of technologies
- Technology packages deal with important areas of open and closed-loop control technology
- Function packages explain the basic functions of automated systems
- Application packages provide basic and further training closely oriented to everyday industrial practice

The technology packages encompass pneumatics, electro-pneumatics, control pneumatics, programmable logic controllers, hydraulics, electro-hydraulics, proportional hydraulics and control hydraulics.

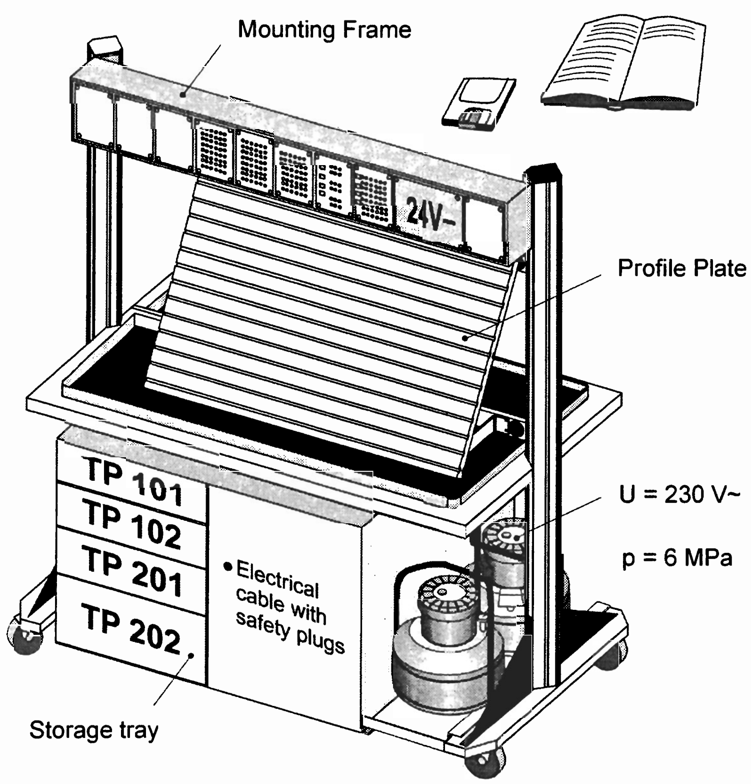


Fig. 1:
Pneumatik 2000 –
i.e. mobile workstation

The modular design of the learning system permits applications beyond the limits of the individual packages. PLC actuation, for example, is therefore possible of pneumatic, hydraulic and electrical actuators.

All learning packages have an identical structure:

- Hardware
- Teachware
- Software
- Courses

The hardware consists of industrial components and installations adapted for didactic purposes.

The courseware is matched methodologically and didactically to the training hardware. The courseware comprises:

- Textbooks (with exercises and examples)
- Workbooks (with practical exercises, worksheets, supplementary notes, solutions and data sheets)
- Overhead transparencies and videos (as a visual means of teaching support)

The teaching and learning media are available in several languages. They have been designed for use in classroom teaching, but can also be used for self-study purposes.

In the software field, computer-based training programs, computer simulating programs, CAD programs and programming software for programmable logic controllers are available.

Festo's Didactic range of products for basic and further training is completed by a comprehensive selection of courses matched to the contents of the technology packages.

Table of contents

Section A – Course

Components of a programmable logic controller

Programming to EN 61131 (IEC 61131)

Basic logic operations

Logic control systems without latching properties

Logic control systems with latching properties

Logic control systems with time response

Sequence control systems

Section B – Fundamentals

Section C – Solutions

Section D – Data sheets

Technology package TP301

“ Programmable logic controllers“

The technology package TP301 "Programmable logic controllers" is a component part of the Festo Didactic Learning System for Automation and Technology and forms the basic level of TP300.

The training aims of TP301 are to learn how to program programmable logic controllers and to teach the fundamentals for creating programs in the programming languages 'ladder diagram' (LD), 'function block diagram' (FBD), 'instruction list' (IL), 'structured text' (ST) and 'sequential function chart' (SFC). Programming is effected in accordance with EN 61131-3 (IEC 61131-3).

You have the option of using this workbook in conjunction with alternative programmable logic controllers by different manufacturers.

A basic knowledge of electro-pneumatics and sensor technology is recommended to work through technology package TP301.

The exercises in TP301 deal with the following main topics:

- Components of a programmable logic controller
- PLC programming to EN 61131 (IEC 61131)
- Basic logic operations
- Logic control systems
- Sequence control systems

The allocation of components and exercises can be seen from the component/exercise table.

Layout of this workbook

The workbook is structured as follows:

Section A – Course

Section B – Fundamentals

Section C – Solutions

Section D – Appendix

Section A – Course teaches the programming of programmable logic controllers with the help of a series of progressive exercises.

Any necessary technical knowledge required for the implementation of an exercise is provided at the beginning. Functions are limited to the most elementary requirements. More detailed knowledge may be gained in section B.

Section C – Solutions provides the solutions to the exercises with brief explanations.

Section B – Fundamentals contains generally applicable technical knowledge to supplement the training contents of the exercises in Section A. Theoretical links are established and the necessary technical terminology explained with the help of examples. An index provides an easy means of locating terminology.

Section D – Appendix which contains data sheets of the used components.

Allocation of component and exercise

Description	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*
Signal input, electrical			1	1	1	1		1	1	1	1	1	1	1		1	1
Signalling device and distributor, electrical			1	1		1			1	1							
Proximity sensor, optical							1			1					1	1	1
Proximity sensor, inductive					1		1				1		1				1
Proximity sensor, capacitive							1				1		1			1	1
Proximity sensor with cylinder mounting												1		4	4	4	4
5/2-way single solenoid valve					1		1	2			1	1	1	1	2	2	2
5/2-way double solenoid valve														1		1	1
Double-acting cylinder					1		1	2						1	2	2	2
Single-acting cylinder											1	1	1	1		1	1
On/off valve with filter regulator valve					1		1	1			1	1	1	1	1	1	1
Manifold					1		1	1			1	1	1	1	1	1	1

Equipment set TP301

Equipment set TP301
Order No.:167101

Description	*Order No.*	*Quantity*
Plastic tubing	151496	
Manifold	152869	1
Single-acting cylinder	152887	1
Double-acting cylinder	152888	2
On/off valve with filter regulator valve	152894	1
Quick push-pull distributor	153128	1
Signal input, electrical	162242	1
Signalling device and distributor, electrical	162244	1
Proximity sensor with cylinder mounting	167060	4
5/2-way single solenoid valve	167074	2
5/2-way double solenoid valve	167076	1
Proximity sensor, inductive	178574	1
Proximity sensor, capacitive	178575	1
Proximity sensor, optical	178577	1

*optional, **not** included in scope of delivery of equipment set*	*Order No.*	*Quantity*
I/O data cable, digital	034031	
Plug-in adapter	035651	
Universal connection unit	162231	
Power supply unit	162416	
Set of cables	167091	

Notes on safety

The following notes should be followed in the interest of safety:

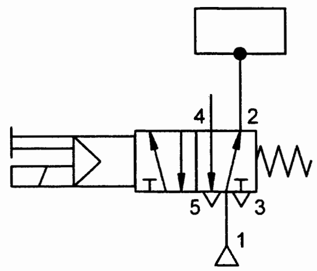
- Mount all components securely on the board.
- Do not switch on compressed air until all line connections have been established and secured.
- Proceed with care when switching on the compressed air.
  Cylinders may advance or retract as soon as the compressed air is switched on.
- Switch off air supply immediately if air lines become detached. This prevents accidents.
- Do not disconnect air lines under pressure.
- Do not exceed the permitted working pressure of 8 bar (800kPa).
- Observe general safety regulations in accordance with EN 60204-1 (IEC 60204-1).
- Use only extra-low voltages of up to 24 V DC.
- Observe the data sheets referring to the individual components, in particular all notes regarding safety.

Operating notes

The following rules should be observed when constructing a circuit:

- Block output 2 of the valve, if a single-acting cylinder is actuated by a 5/2-way single solenoid valve in a circuit.

Fig. 2: Plug for output 2 of a 5/2-way valve



- Input signals, which would result from an actual production process sequence, are reproduced in part by signals via push buttons or switches.

Section A – Course

Logic control systems with latching properties

Logic control systems with time response

Sequence control systems

Subject

Programmable logic controllers

Title

Design and commissioning of a programmable logic controller

Components of a PLC

Training aim

- To be able to explain the basic design and mode of operation of a PLC
- To be able to configure and commission a PLC

Technical knowledge

Nowadays, programmable logic controllers form part of any automation process. Fig. A1.1 illustrates the typical configuration of an automation solution realised by means of a PLC. The control system shown represents the simpler, non-networked group of PLC applications.

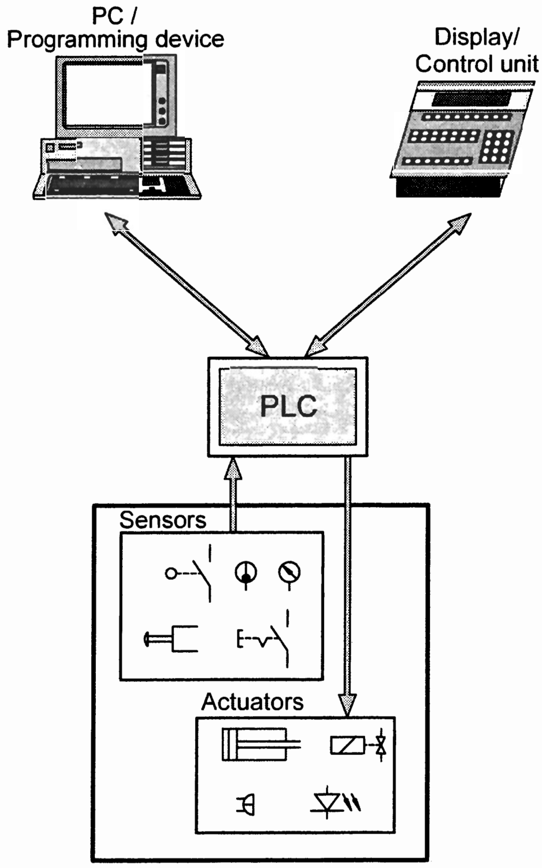


Fig. A1.1:
Automation via PLC

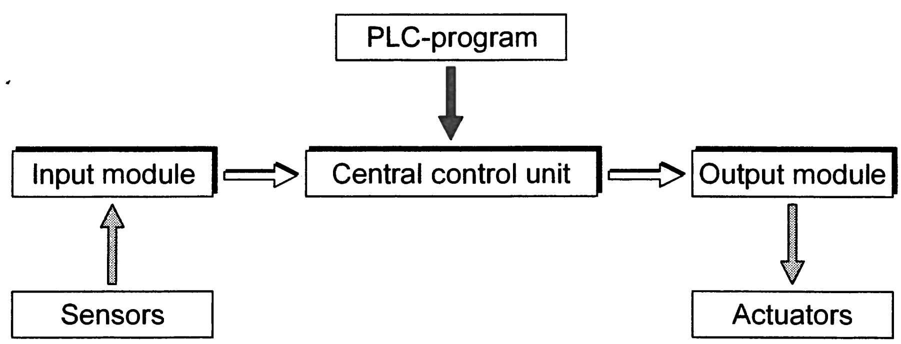
The basic components of the control system are:

- Programmable logic controller (PLC)
  By this, we understand the electronic modules through which all of the system or machine functions to be controlled are addressed and activated in a logic sequence.
- Sensors
  These components are located directly on the system or machinery be controlled, through to which the PLC is communicated actual statuses.
- Actuators
  These components are located directly on the system or machinery to be controlled, through which the PLC is able to change or influence statuses and as such the technical process.
- PC or programming device
  This is used to create the program containing the logic of the system or machinery to be controlled and to transfer this to the memory of the PLC. At the same time, these programming tools also provide supporting functions for the testing of the PLC program and commissioning of the controller.
- Display and control units
  These enables you to monitor and influence the operation of the system or machinery.

Programmable logic controller

The most important component of a control system is the PLC and its program. Fig. A1.2 illustrates the system components of a PLC.

Fig. A1.2: System components of a PLC



A PLC is connected to the system to be controlled via input and output modules. The system to be controlled supplies input signals (mostly binary) via sensors to the input modules. These signals are processed within the main processing unit, the main component of the PLC. Prior .to formulation of IEC standards, known as "central control unit" (CCU). The "specification" for the processing of signals is defined in the PLC program. The result of the processing is output to the actuators of the system to be controlled via the output module. Thus, the design of a PLC corresponds to that of a computer.

PLC program

PLC programs consist of a logic sequence of instructions. The control program is stored in a special, electronic readable memory, the so-called program memory of the PLC. Special RAMs with back-up battery are used during the program development, since its contents can always be changed again very quickly.

After commissioning and error-free function of the controller it is a good idea to transfer the PLC program unerasably to a read-only memory, e.g. an EPROM. If the program is executed, it will be processed in continuous cycles.

Signals

Input signals reach the PLC via sensors. These signals contain information about the status of the system to be controlled. It is possible to input binary, digital and analogue signals.

A PLC can only recognise and output electrical signals. For this reason, non-electrical signals are converted into electrical signals by the sensors. Sensor examples are:

- Push buttons, switches, limit switches, proximity sensors

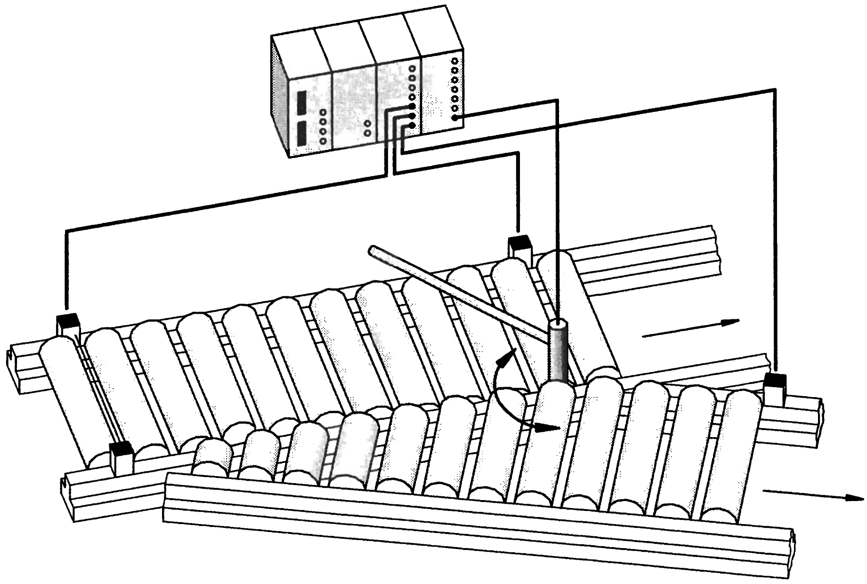
Output signals influence the system to be controlled. The signals can be output in the form of binary, digital or analogue signals. Output signals are amplified into switching signals via the actuators or converted into signals of other energy forms. Actuators examples are:

- Lamps, buzzers, bells, contactors, cylinders with solenoid valves, stepper motors

Problem description

A control task is to be solved via a programmable logic controller (PLC). Familiarise yourself with the basic design of a PLC.

Positional sketch



Exercise definition

1. Components of a PLC
2. Design and commissioning of the PLC you have selected

Implementation

To carry out the exercise using the worksheets, refer to Section B of the workbook and your PLC data sheet or manual.

WORKSHEET

1.1 Components of a PLC

Question 1:
What are the basic components of a programmable logic controller?

Question 2:
What are the basic modules making up the central control unit of a programmable logic controller?

Question 3:
How is electrical isolation achieved between sensor/actuator signals and the PLC?

WORKSHEET

1.2 Design and commissioning of the PLC you have selected

Enter the technical data of the selected programmable logic controller in the table below.

Criteria	*Technical data*
Operating voltage	
Nominal voltage	
Permissible voltage range	
Current consumption	
Inputs	
Number	
Input current	
Input level	
Outputs	
Number	
Switching logic	
Output voltage	
Output current	

Technical data

Configure the PLC in accordance with the notes in the relevant data sheet or manual.

Subject

Programmable logic controllers

Title

From problem to solution – taking into consideration EN 61131 (IEC 61131)
Practical steps for PLC programming

Training aim

- To familiarise yourself with the basic language resources for the configuration and structuring of a PLC program in accordance with EN 61131 (IEC 61131).
- To be able to declare variables for use in a PLC program
- To be able to apply a systematic procedure for the implementation of PLC exercises

Technical knowledge

Creating a PLC program

The practical steps for creating a PLC program are illustrated in Fig. A2.1.

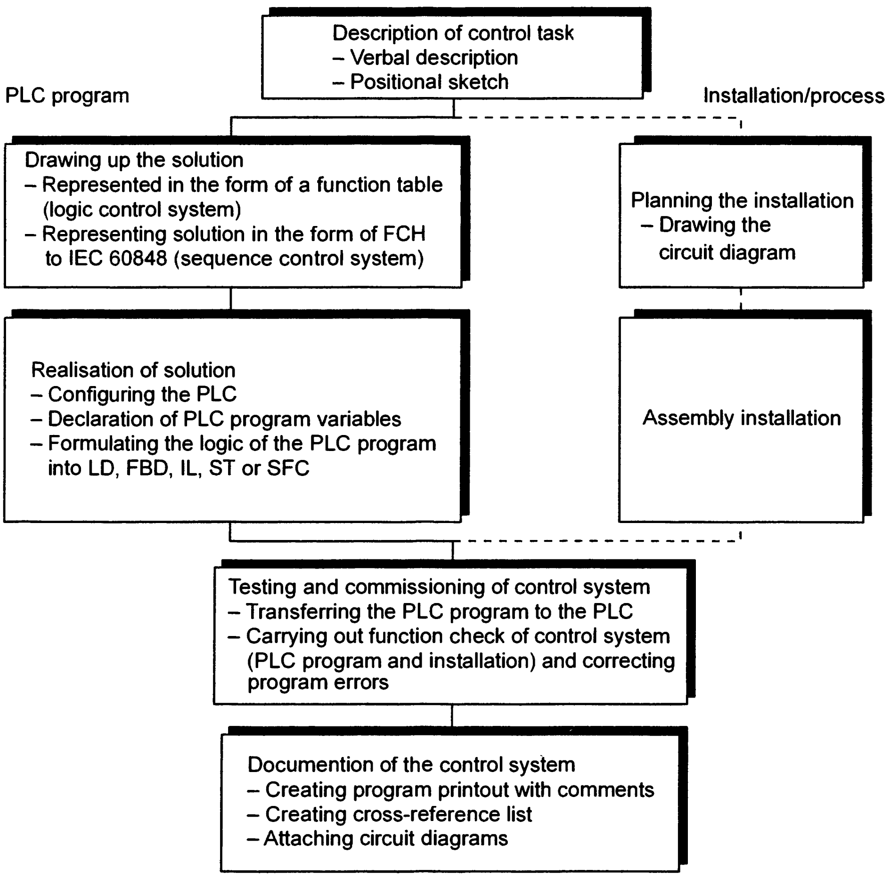


Fig. A2.1: Procedure for creating a PLC program

Description of a control task

The basic requirements for describing a control task include a verbal description, a positional sketch and the definition of sensors and actuators to be used. This information is required for the development of circuit diagrams. Moreover, it is necessary to define the allocation of sensors to the PLC inputs or the allocation of actuators to PLC outputs. An example of an allocation list of this type is shown in table A2.1.

Table A2.1: Example of an allocation list

Resource designation	*Input/output address on PLC*	*Comment*
S1	I1.5	Push button START
3Y2	Q2.7	Cylinder 3A1 advance

Since this list forms a component part of PLC programs, it is not created separately in the case of smaller control tasks.

Solution design

The designed solution is to give a clear representation of the function and behaviour of the controller independent of technology. The function table is used as a means of describing simple logic control systems. The function chart to IEC 60848 is particularly suitable for the description of sequence control systems.

Realisation of the solution

The realisation of the solution is divided into

- Programming of the logic of the control system (PLC)
- Incorporating the PLC program in the PLC or PLC system

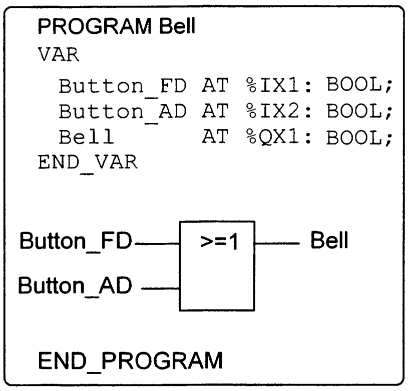
An example of this is shown in Fig. A2.2 using a simple control task:

Example

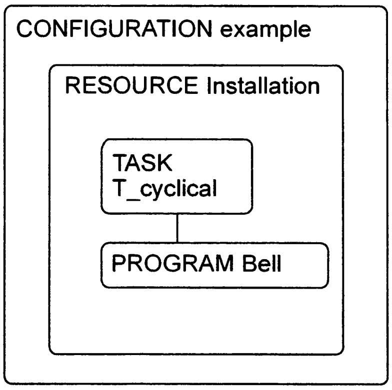
A bell is to ring either if the bell button on the front door or the bell button on the apartment door is actuated.

When commencing the production of a PLC program, the resources are to be declared in the syntax defined by EN 61131-3 (IEC 61131-3). The PLC inputs and outputs declared in Fig. A2.2a are local variables within the program "Bell". The use of global variables is only required for more complex control tasks.

Following the declaration, the program body is formulated. Functions and function blocks are available at this point in order to create a clearly arranged program. The example is programmed in function block diagram using the OR function.



a) Example of a PLC program



b) Example for the installation of a PLC program in a PLC or a PLC system

Fig. A2.2: Realisation of a solution

In order to execute a PLC program via the PLC, it is necessary to know how the program is to be processed: cyclically or in relation to certain events. EN 61131-3 (IEC 61131-3) provides its own language resources for the assignment of such sequence characteristics and for incorporating the program in the PLC or the PLC system. These are the configuration language resources.

Fig. A2.2b illustrates the configuration "Example". This configuration represents the PLC. The configuration "Example" requires the resource "Installation". This resource is assigned to the program "Bell". The task "T_cyclical" defines that the program "Bell" is to be processed cyclically.

Testing and commissioning the controller

The program is loaded from the PC or programming device to the PLC for the testing or commissioning of the control system. Following this, the interaction of PLC and system must be checked.

Control system documentation

The system documentation is compiled as soon as the installation operates free of fault and the PLC program has been corrected accordingly. The documentation basically consists of:

- the positional sketch,
- the formal solution design and
- the program printout with comments.

Problem description

PLC inputs and outputs and additional variables for storing information are to be incorporated in a PLC program. For this, you will need to familiarise yourself with the basic procedure required for PLC program generation.

Exercise definition

1. Procedures for creating a PLC program Exercise definition
2. Resources of a PLC according to EN 61131-3 (IEC 61131-3)
3. Declaration of variables according to EN 61131-3 (IEC 61131-3)

Implementaion

In order to carry out the exercise you will need the information from Section B of the workbook: Chapter 6.

WORKSHEET

2.1 Practical steps for creating a PLC program

Specify the five practical steps for creating a PLC program.

Question:

What activities are carried out in the step "Implementation of the solution"?

2.2 Resources of a PLC in accordance with EN 61131-3 (IEC 61131-3)

The following resources are to be addressed directly. Specify the designations in accordance with EN 61131-3 (IEC 61131-3):

Input bit 14 ________________________________

Memory 9 ________________________________

Output word 3________________________________

Input 7 on 2nd. Input card ________________________

WORKSHEET

2.3 Declaration of variables to EN 61131-3 (IEC 61131-3)

The following data must be taken into consideration in a program declaration. Use the appropriate data type in your declaration. The declaration is to be valid locally only.

- Input for a switch S1, applied to input 2 of the 4th input card
- Temperature TEMP, applied to output word No. 1
- Memory VALVE_OPEN
- boolean memory with identifier PART_PRESENT, preallocated initial value 0
- boolean memory with identifier ROBOT_INIT, preallocated with initial value 1
- storage of a number (INT) under the name NUMBER, preassigned the value 0

Subject

Programmable logic controllers Subject

Title

Lamp circuit
The assignment function

Training aim

- To understand the actuation of a PLC output
- To be able to realise the logic assignment function with a PLC
- To be able to create a PLC program in accordance with EN 61131-3 (IEC 61131-3)

Technical knowledge

Each programmable logic controller has a certain number of inputs and outputs, through which it is connected with the sensors and actuators. The program transferred to the controller contains the commands which interconnect the individual inputs and assigns these to the corresponding outputs.

The assignment function

The assignment function permits a PLC input signal to be directly transmitted to a PLC output. The behaviour can be clearly described with the help of a function table, which represents this for an input %IX1 and an output %QX2 in table A3.1.

%IX1	%QX2
0	0
1	1

Table A3.1: Function table for the assignement function

In order to realise the assignment function in the individual programming languages, you will need the commands shown in table A3.2.

Table A3.2: The assignment function

LD

Current rung with normally open contact and non-stored coil.

FBD

%IX1 ———— %QX2

Direct connection of specified input and output.

IL

```
LD      %IX1
ST      %QX2
```

Reading of value of specified input to accumulator.

Storage of contents of accumulator to specified output.

ST

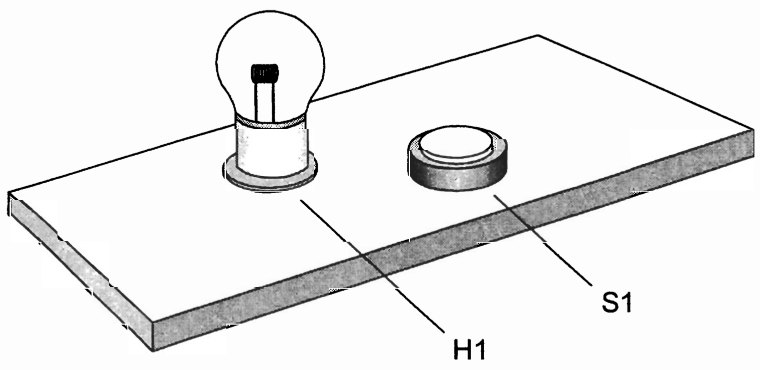
```
%QX2  :=  %IX1;
```

Assignment of value of specified input to the right of " := " to the specified output on the left of " := ".

Problem description

Actuation of a push button (S1) is to cause a lamp (H1) to be switched on. The lamp is to be illuminated as long as the push button is actuated.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of PLC program variables
4. Formulation of the PLC program in the various programming languages
5. Testing and commissioning of the PLC program and system

Implementation

3.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Signal input, electrical
1	Signalling device

- **Prior to wiring:** Switch off power supply!
- Establish the electrical connections.

3.2 Describe the control task by means of the function table and the boolean equation

- Describe the behaviour of the control system irrespective of technology by means of the function table and the associated boolean equation.

3.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

3.4 Formulation of the PLC program into one of the PLC programming languages

- Select one of the programming languages supported by your PLC system. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

3.5 Testing and commissioning of PLC program and system

- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!

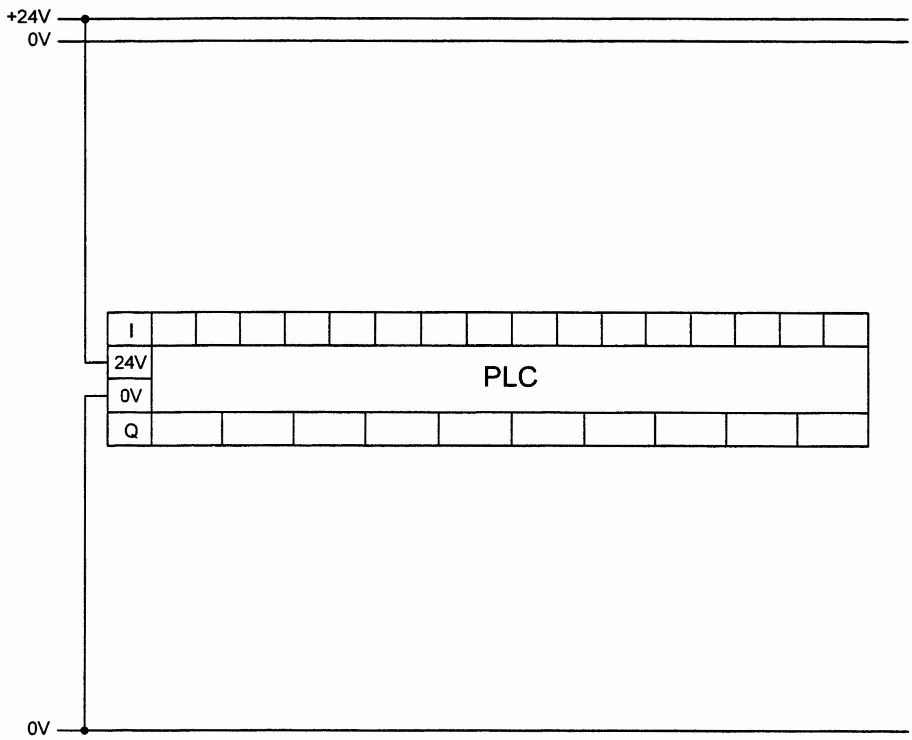
- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

3.1 Drawing up the electrical circuit diagram and constructing the circuit

Complete the electrical circuit diagram and enter the available input and output addresses of your PLC.

Circuit diagram, electrical



3.2 Describing the control task by means of the function table and the boolean equation

Create the function table

Function table

S1	H1
0	
1	

Derive the boolean equation from this:

Booleam equation

3.3 Declaration of PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

3.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task in one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

Worksheet

PLC program

Question 1:

What is the behaviour of a non-stored programmed output, if the input signal is no longer applied?

Question 2:

Describe the basic design of an output module.

Subject

Programmable logic controllers

Title

Burglar alarm
The NOT function

Training aim

- To be able to realise the logic NOT function with a PLC Training aim

Technical knowledge

The NOT function

The NOT function is used to convert binary signals into opposing signals:

- If the signal is 0, it is evaluated as 1
- If the signal is 1, it is evaluated as 0

Table A4.1 illustrates an example of the behaviour of the NOT function or an input %IX1 and an output %QX2.

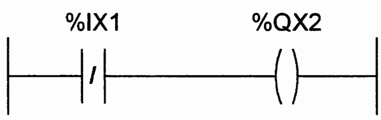
%IX1	*%QX2*
0	1
1	0

Table 4.1: Function table for NOT function

Table A4.2 contains the commands for the implementation of the NOT function in the individual programming languages.

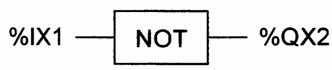
Table A4.2: The NOT function

LD



Current rung with normally closed contact and non-stored coil.

FBD



Use NOT function at specified input.

IL

`LDN   %IX1`	Reading of negated value of specified input to accumulator.
`ST    %QX2`	Storage of contents of accumulator to specified output.

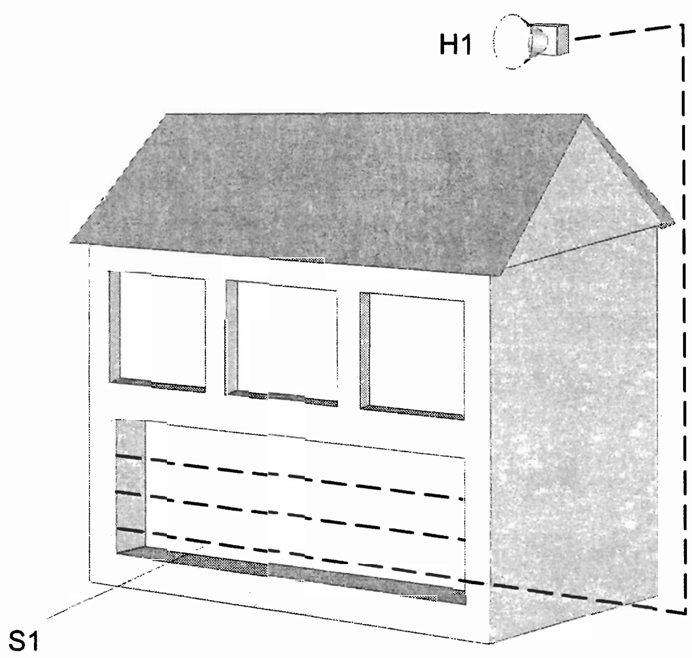
ST

`%QX2  :=  NOT  %IX1;`	Assignment of negated value of specified input to the output on the left of " := ".

Problem description

A thin wire has been stretched behind a display window, which breaks if a burglary is attempted. A closed circuit is interrupted as a result of this and a buzzer is sounded.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of the PLC program variables
4. Formulation of the PLC program into one of the PLC programming languages
5. Testing and commissioning of the PLC program and system

Implementation

4.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted profile plate:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Signal input, electrical
1	Signalling device

- **Prior to wiring:** Switch of power supply!
- Establish the electrical connections.

4.2 Describing the control task by means of the function table and the boolean equation

- Describe the behaviour of the control system irrespective of technology by means of a function table and the associated boolean equation.

4.3 Declaration of PLC program variables

- All variables must be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

4.4 Formulation of the PLC program into one of the PLC programming languages

- Select one of the programming languages supported by your PLC system. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

4.5 Testing and commissioning of the PLC program and system

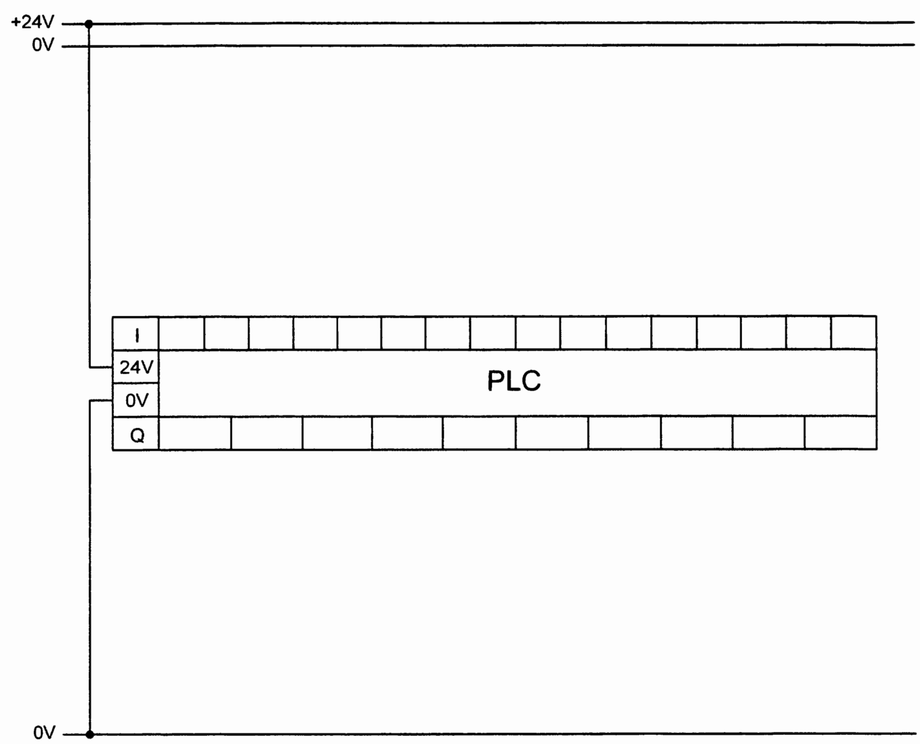
- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

4.1 Drawing up the electrical circuit diagram and constructing the circuit

Complete the electrical circuit diagram and enter the available input and output addresses for your PLC.



Circuit diagram, electrical

4.2 Describing the control task by means of the function table and the boolean equation

Create the function table:

S1	*H1*
0	
1	

Function table

Derive the boolean equation from this:

Boolean equation

4.3 Declaration of the PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

4.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task in one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

WORKSHEET

PLC-program

Question:

The output is set non-stored. What is the effect on the output signal, if the wire is broken and has to be repaired?

Subject

Programmable logic controllers

Title

Press with protective guard
The AND function

Training aim

- To be able to realise a logic AND function with a PLC
- To understand the term 'function' according to EN 61131-3 (IEC 61131-3)
- To be able to use standard functions to EN 61131-3 (IEC 61131-3)

Technical knowledge

Functions are part of the program organisation units and therefore represent a means for configuring PLC programs. EN 61131-3 (IEC 61131-3) provides standardised functions for the solution of basic control technology tasks.

The AND function

Only when all AND connected signals are 1, is the result 1. If one of the connected signals is 0, then the result is also 0.

The function table for the AND function is illustrated below for the two inputs %IX1 and %IX2 as well as %QX3.

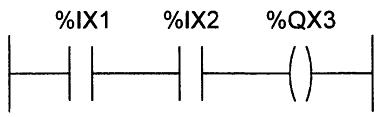
Table A5.1: Function table for the AND function

%IX1	%IX2	%QX3
0	0	0
0	1	0
1	0	0
1	1	1

Table A5.2 contains the commands for the realisation of the AND function in the individual programming languages.

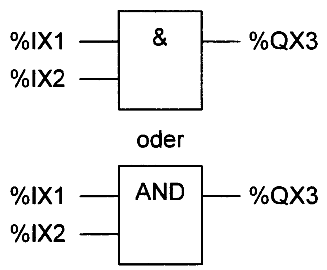
Table A5.2: The AND function

LD



Rung with consecutively switching normally open contacts

FBD



The inputs and the output of the AND function are connected with the specified current parameters.

IL

`LD %IX1`	Reading the value of the specified input to the accumulator.
`AND %IX2`	AND connection of current result with second input.
`ST %QX3`	Storage of contents of accumulator to specified output.

ST

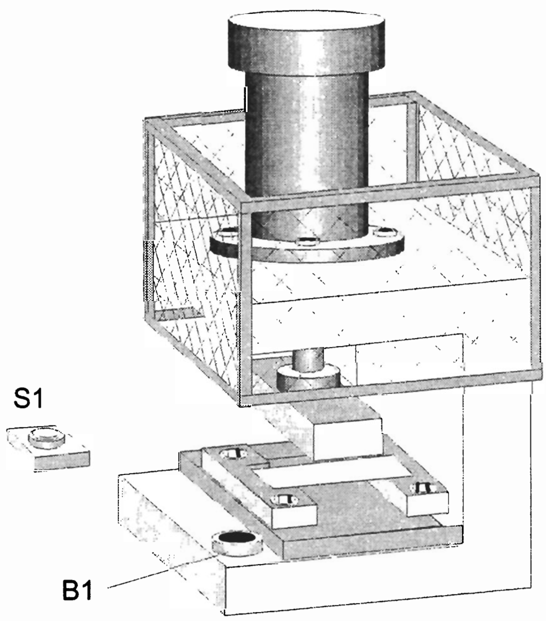
`%QX3 := %IX1 & %IX2;`	The two specified inputs are connected with the "&" operator. The result is assigned to the specified output.
or	
`%QX3 := AND(%IX1, %IX2);`	Invocation of AND function with specified inputs as current transfer parameters.

Problem description

A press stamp 1A1 is to advance only if a push button S1 is actuated and a protective guard is closed. If one of these conditions is not met, the press tool is to return immediately.

The position of the closed protective guard B1 is monitored by a proximity switch B1. The press tool is advanced or retracted by means of a spring return solenoid valve (coil 1Y1).

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of the PLC program variables
4. Formulation of the PLC program into one of the PLC programming languages
5. Testing and commissioning of the PLC program and system

Implementation

5.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted profile plate:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Double-acting cylinder
1	5/2-way single solenoid valve
1	Signal input, electrical
1	Proximity switch
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air supply at service!

- Establish the electrical and pneumatic connections.

5.2 Describing the control task by means of the function table and the boolean equation

- Describe the behaviour of the control system irrespective of technology by means of a function table and the associated boolean equation.

5.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration, which are required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC program system used.

5.4 Formulation of the PLC program into one of the PLC programming languages

- Select one of the languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

5.5 Testing and commissioning of PLC program and system

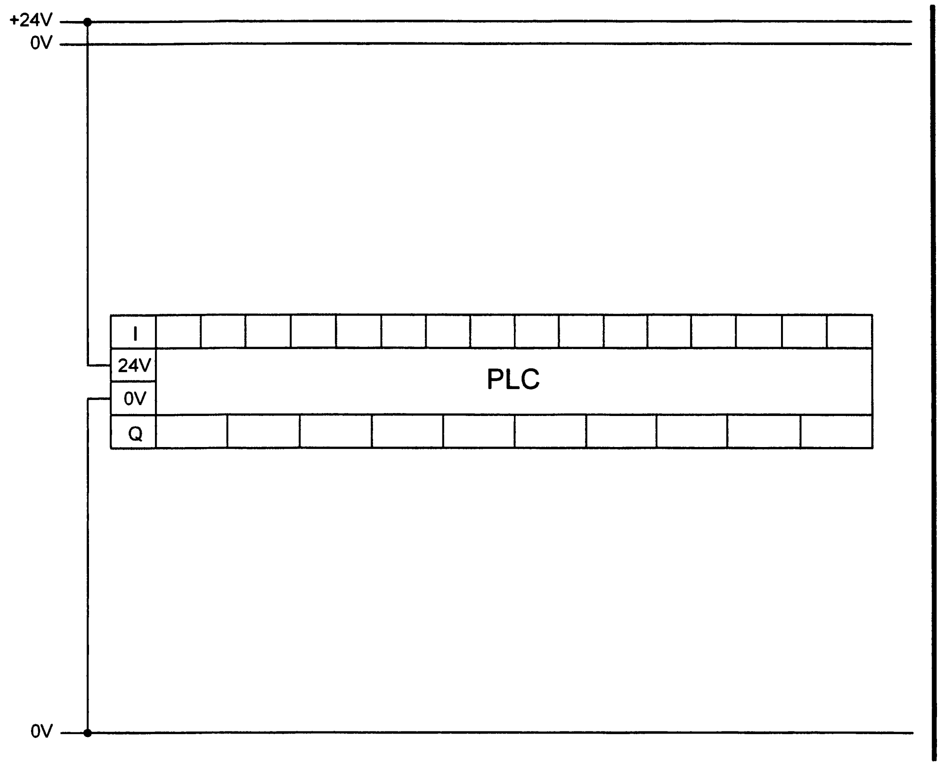
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply at the service unit to operating pressure (see data sheets for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

5.1 Drawing up the electro-pneumatic and the electrical circuit diagram and constructing the circuit

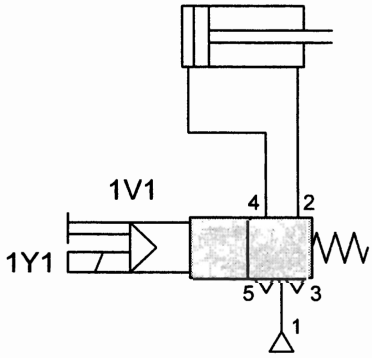
Complete the electrical circuit diagram and enter the available input and output addresses for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



5.2 Describing the control task by means of the function table and the boolean equation

Create the function table

Function table

S1	*B1*	*1Y1*

Derive the boolean equation from this:

Boolean equation ______________________________

WORKSHEET

5.3 Declaration of the PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

5.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC-Program

Question:

Does output 1Y1 have to be set stored or non-stored?

Subject

Programmable logic controllers

Title

Bell system
The OR function

Training aim

- Realising the logic OR function with a PLC.

Technical knowledge

The OR function

If at least one of the connected signals is 1, the result is also 1. Only if all the connected signals are 0 is the result also 0.

Table A6.1 contains the function table for the OR connection of the signal from input %IX1 and input %IX2. The result is mapped to output %QX3.

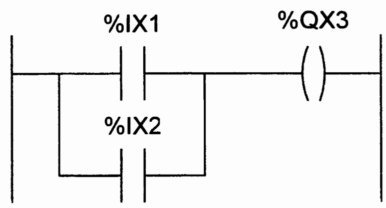
%IX1	*%IX2*	*%QX3*
0	0	0
0	1	1
1	0	1
1	1	1

Table A6.1: Function table for the OR function

Table A6.2 contains the commands for the realisation of the OR function in the individual programming languages.

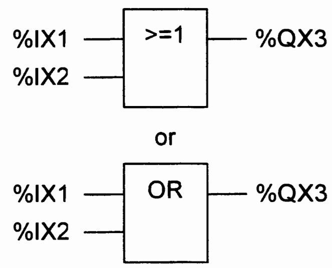
Table A6.2: The OR function

LD



Rung with parallel switching normally open contacts.

FBD



The inputs and the output of the OR function are connected with the current parameters specified.

IL

`LD %IX1`	Reading the value of the specified input to the accumulator.
`OR %IX2`	OR connection of current result withsecond input.
`ST %QX3`	Storage of contents of accumulator to specified output.

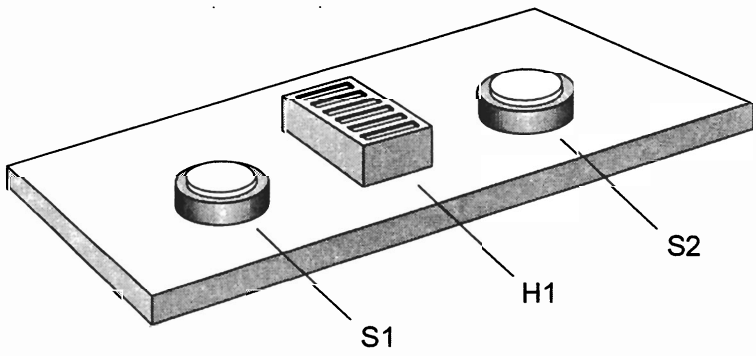
ST

`%QX3 := %IX1 OR %IX2;`	The two specified outputs are connected with the operator "OR". The result is assigned to the specified output.

Problem description

An apartment bell is to ring if bell button S1 at the front door is pressed or bell button S2 at the apartment door.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of the PLC program variables
4. Formulation of the PLC program into one of the PLC programming languages
5. Testing and commissioning of PLC program and system

Implementation

6.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted profile plate:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Signal input, electrical
1	Signalling device

- **Prior to wiring the installation:**
  Switch off power supply!

- Establish the electrical connection.

6.2 Describing the control task by means of the function table and the boolean equation

- Describe the behaviour of the control system irrespective of technology by means of a function table and the associated boolean equation.

6.3 Declaration of PLC program variables

- All variables must be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

6.4 Formulation of the PLC program into one of the PLC programming languages

- Select one of the programming languages supported by your PLC system. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

6.5 Testing and commissioning of the PLC program and system

- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!

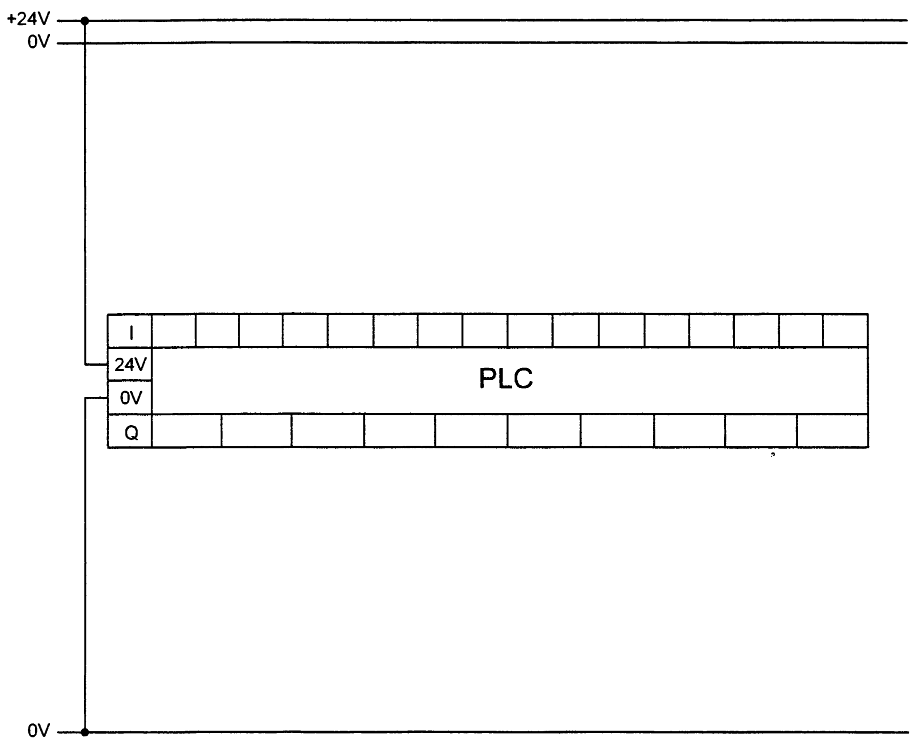
- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

6.1 Drawing up the electrical circuit diagram and constructing the circuit

Complete the electrical circuit diagram and enter the available input and output addresses for your PLC.

Circuit diagram, electrical



6.2 Describing the control task by means of the function table and the boolean equation

Create the function table:

Function table

S1	S2	H1

Derive the boolean equation from this:

Boolean equation ______________________________

6.3 Declaration of the PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	Data type	Address	Comment

6.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task in one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

WORKSHEET

PLC program

Question:

In the case of the OR function, the output is set if at least one input signal is set. With the exclusive OR unction, the output is set only if exactly one of the connected inputs is set. Completes the function table.

Function table

S1	*S2*	*H1*
0	0	
0	1	
1	0	
1	1	

Subject

Programmable logic controllers

Title

Stamping device
Combinations of AND/OR/NOT

Training aim

- To be able to realise combinations of logic connections with a PLC
- To understand the priorities of elementary operators in the individual programming languages

Technical knowledge

Combination of logic connections

Many control tasks require the programming of a combination of logic connections. The following are essential for drawing up a solution:

- Establishing a boolean equation which describes the logic of the control task
- Taking into consideration the priorities of the operators used for programming

The example below deals with a combination of AND, OR and NOT functions:

A lamp H1 is to illuminate if switch S1 and, in addition exactly one of the switches S2 or S3 is actuated.

The relevant function table is as follows:

Table A7.1: Function table

S1	*S2*	*S3*	H1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

The disjunctive normal form (DNF) can be derived from this table. The DNF describes the control task by means of a boolean equation, which can be easily converted into one of the programming languages.

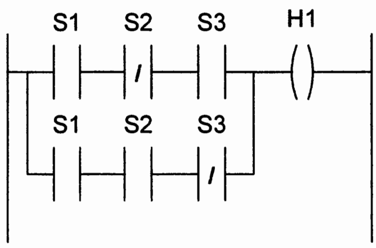
The solution method for the DNF is based on the lines in the function table, for which the result variable (H1) has the value 1. These lines are OR connected and lead to the boolean equation

$$H1 = (S1 \wedge \overline{S2} \wedge S3) \vee (S1 \wedge S2 \wedge \overline{S3})$$

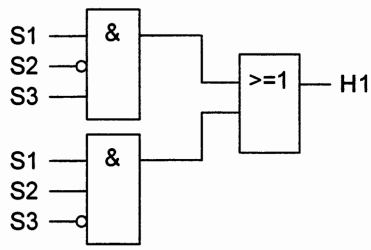
The realisation of the control task in the individual programming languages is listed in table A7.2.

Table A7.2: Combination of logic operation

LD



FBD



IL

```
LD      S1
ANDN    S2
AND     S3
OR(     S1
AND     S2
ANDN    S3
)
ST      H1
```

ST

```
H1 := S1 & NOT S2 & S3 OR S1 & S2 & NOT S3;

or

H1 := (S1 & NOT S2 & S3) OR (S1 & S2 & NOT S3);
```

Since the processing of the OR command is a subsequent action in the statement list, an opening parenthesis follows the OR command. The closing parenthesis in the penultimate line causes the result of the parenthesized expression to be OR connected with the current result (in the accumulator).

The solution in Structured Text does not require any parenthesising since the operators already have priorities. In order to obtain better readability it is however a good idea to use parentheses for more complex expressions.

Priorities with basic logic connections

In the graphic programming languages **LD** and **FBD** the order of processing is implicitly specified by the graphics of the program or program part.

In this way, a series connection is evaluated first in a current rung before a simultaneously existing connection is "calculated".

In the function block diagram, the order of evaluation of a network is defined by blocks.

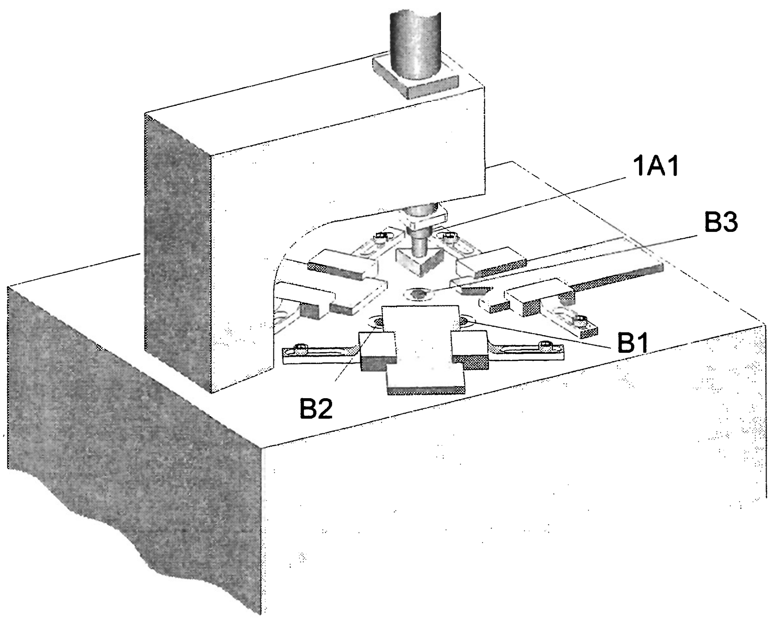
The instructions of an **IL** are processed line by line and the operators therefore all have the same priority. If the evaluation of an operator is to be a subsequent action, this must also be done by means of parenthesis.

The language **ST** defines a unique rule of precedence for the operators. The boolean AND has a higher priority than the boolean OR. With mathematical operations, the rule multiplication before addition or subtraction applies.

Problem description

A stamping device can be operated from three sides. A workpiece is inserted via a guide, whereby it touches two of the three proximity switches B1, B2 and B3. This causes a pneumatic cylinder 1A1 to extend via a solenoid valve (coil 1Y1), whereby a recess is to be stamped into the workpiece. The stamping cycle is to be triggered only if two signal generators are addressed. For reasons of safety the cylinder must be prevented from advancing, if all three proximity sensors are contacted.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of the PLC program variables
4. Formulation of the PLC program into one of the PLC programming languages
5. Testing and commissioning of the PLC program and system

Implementation

7.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Double-acting cylinder
1	5/2-way single solenoid valve
1	Proximity switch, inductive
1	Proximity switch, capacitive
1	Proximity switch, optical
1	Plastic tubing

- **Prior to wiring and tubing of the installation:**
  Switch off power supply!
  Switch off air supply at service unit!

⚠

- Establish the electrical and pneumatic connections.

7.2. Describing the control task by means of the function table and the boolean equation

- Describe the behaviour of the control system irrespective of technology by means of a function table and the associated boolean equation.

7.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

7.4 Formulation of PLC program into one of the PLC programming languages

- Select one of the languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

7.5 Testing and commissioning of PLC program and system

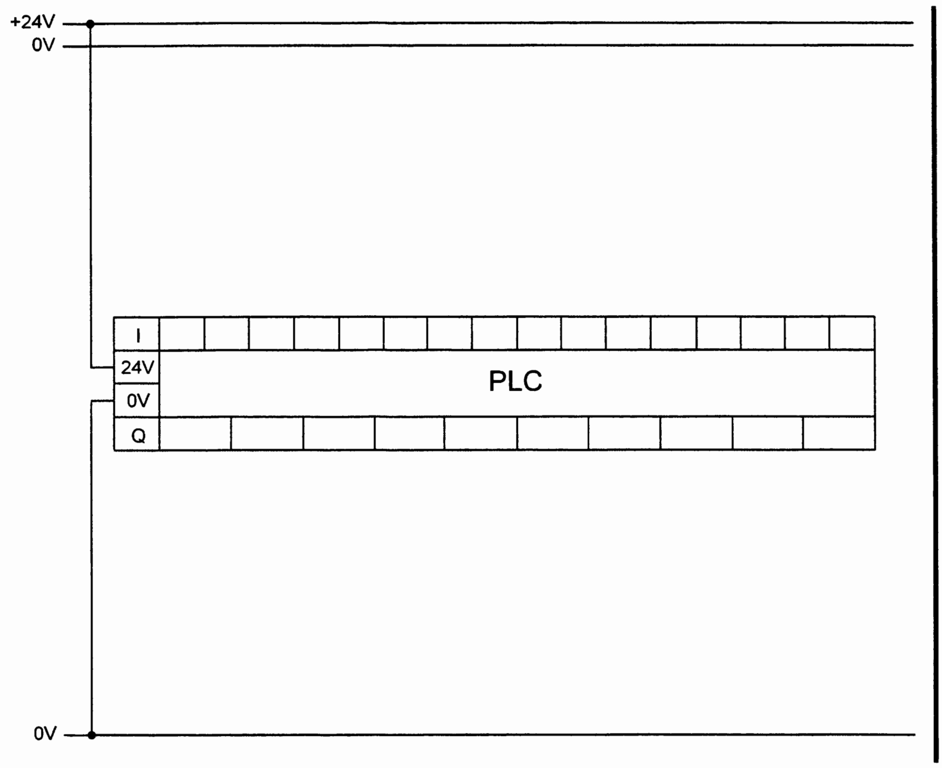
- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC! Increase air supply on service unit to operating pressure (see data sheets of pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document the solution.

WORKSHEET

7.1 Drawing up the electrical circuit diagram and constructing the circuit

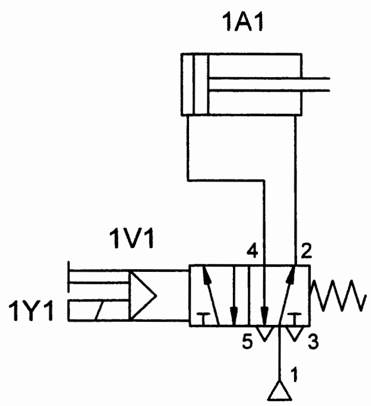
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system

Circuit diagram, electro-pneumatic



7.2 Describing the control task by means of the function table and the boolean equation

Complete the function table:

Function table

B1	*B2*	*B3*	1Y1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

State the associated boolean equation:

Boolean equation ______________________________

WORKSHEET

7.3 Declaration of variables of the PLC program

Declare the variables required in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

7.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Question:

Why does the negated element have to occur in each parenthesis?

Subject

Programmable logic controllers

Title

Silo control system for two bulk materials
Logic control system with branching

Training aim

- To be able to solve a logic control system with branching

Technical knowledge

More than one final control element is addressed even in the case of simple control tasks.

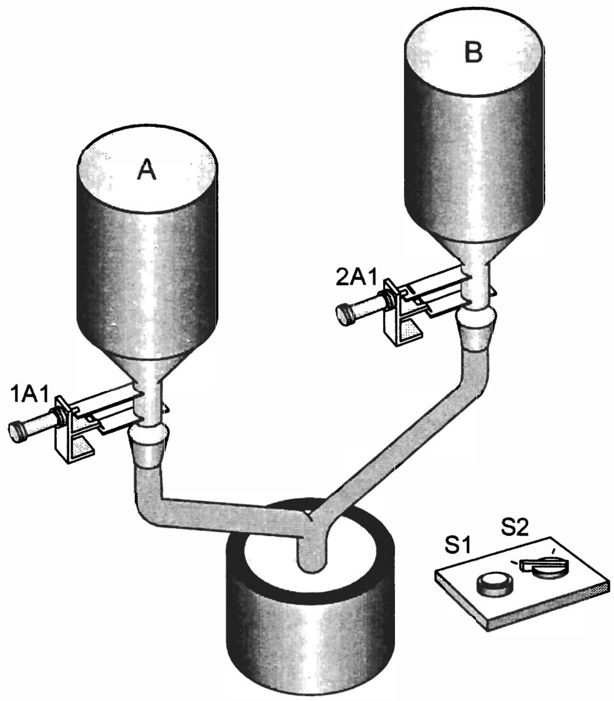
This requires the PLC to actuate not just one but several outputs. In the graphic languages this leads to the programming of several current rungs or networks. These networks or current rungs may be optionally provided with a network identifier. A network identifier is required if a jump is to be executed to this network within the program.

Problem description

A mixing plant permits a selection between two bulk materials per selector switch (S2). In switch position 1 (S2 = 0 signal), bulk material A reaches a mixing container, if push button S1 is actuated simultaneously.

Similarly, bulk material is conveyed, if selector switch S2 is in position 2 (S2 = 1 signal) and push button S1 is actuated. Silo A is opened via cylinder 1A1 (solenoid valve 1Y1), Silo B via cylinder 2A1 (solenoid valve 2Y1).

Positional sketch



Exercise definition

1. Drawing up the circuit diagram and constructing the circuit
2. Describing the control task by means of the function table and the boolean equation
3. Declaration of the PLC program variables
4. Formulation of the PLC program into one of the PLC programming languages
5. Testing and commissioning of PLC program and system

8.1 Drawing up the circuit diagram and constructing the circuit

Implementation

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required components on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
2	Double-acting cylinder
2	5/2-way single solenoid valve
1	Signal input, electrical
	Plastic tubing

- **Prior to wiring and tubing of the installation:**
  Switch off power supply!
  Switch off air supply at service unit!

- Establish the electrical and pneumatic connections.

8.2 Describing the control task by means of the function table and the boolean equation

- Describe the behaviour of the controller irrespective of technology by means of a function table and the associated boolean equation.

8.3 Declaration of the PLC program variables

- All variables must be created as program-local variables.
- Specify only those parts of the declaration required or your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

8.4 Formulation of the PLC program into one of the PLC programming languages

- Select one of languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

8.5 Testing and commissioning of the PLC program and system

- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using standard voltage of 24 V DC! Increase air supply on service unit to operating pressure (see data sheets for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

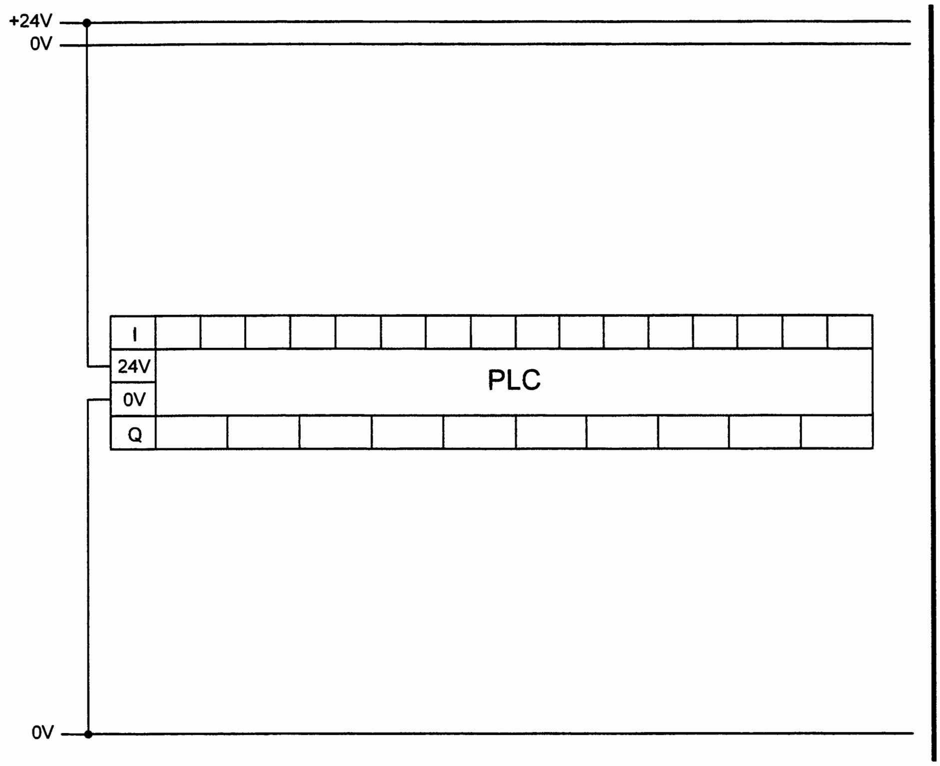
- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

8.1 Drawing up the electrical circuit diagram and constructing the circuit

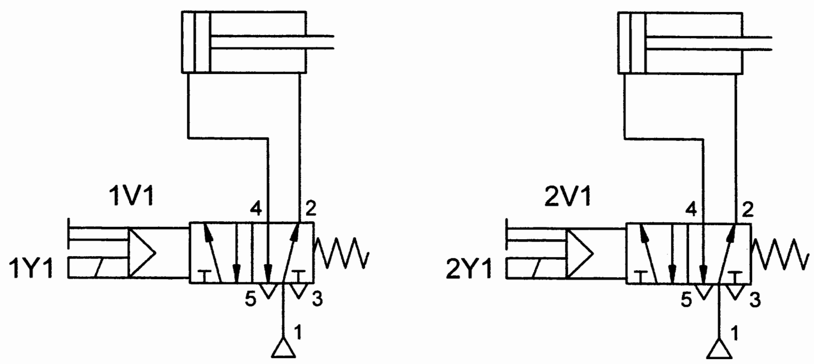
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.

Ciruit diagram, electrical



Configure the control system

Circuit diagramm, electro-pneumatic



8.2 Describing the control task by means of the function table and the boolean equation

Complete the function table:

Function table

S1	*S2*	*1Y1*	2Y1
0	0		
0	1		
1	0		
1	1		

State the associated boolean equation:

Boolean equation ____________________

WORKSHEET

8.3 Declaration of the PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

8.4 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Subject
Programmable logic controllers

Title
Fire alarm
Setting an output

Training aim
- To be able to set and store an output of a PLC
- To be able to understand function blocks to EN 61131 (IEC 61131)
- To be able to use the standard function blocks SR flip-flop and RS flip-flop

Technical knowledge
Storage operations form part of the elementary PLC operations and apply in cases where a briefly occurring signal status is permanently stored. Typical examples of storage operations are the permanent setting or resetting of output signals. The standard function blocks SR and RS are available for the realisation of storage functions.

Function block SR, dominant setting flip-flop

Function block SR (Fig. A9.1) contains a dominant setting flip-flop.

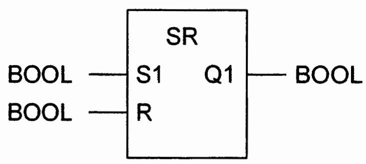


Fig. A9.1:
Function block SR

The typical behaviour of the SR function block is therefore as follows:

- A 1-signal at set input S1 sets the flip-flop, i.e. the value of Q1 becomes 1.
- A 1-signal at reset input R sets the value of Q1 at 0 only if a 0-signal simultaneously applies at the S1 input.
- If a 1-signal applies both at the S1 and the R input output Q1 is set.

Realisation of storage function "Set" in the individual programming languages

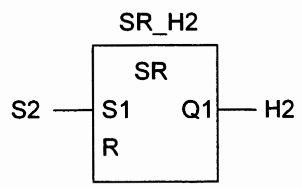
The solutions for the following example are listed in table A9.1.

Example Lamp H2 is to be illuminated via actuation of a push button S2.

Table A9.1: The storage function Set

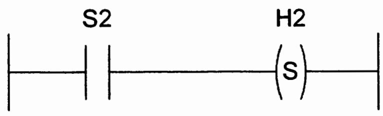
```
VAR
    S2  AT  %IX3    :  BOOL;    (* Push button S2 at input IX3 *)
    H2  AT  %QX4    :  BOOL;    (* Lamp H2 at output QX4       *)
    SR_H2           :  SR;      (* SR-flip-flop named SR_H2    *)
                                (* For storage of status       *)
                                (* of H2                       *)
END_VAR
```

FBD



SR flip-flop SR_H2 for storage of status of variable H2.
Direct assignment of value of Q1 at variable H2.

LD



Rung with normally open contact and set coil.

IL

```
LD    S2                   Reading of value of S2
S     H2                   Setting of the variable H2
or
CAL   SR_H2 (S1  :=  S2)   Invocation of flip-flop SR_H2 with
                           current transfer parameter S2.
LD    SR_H2.Q1             Reading of output value Q1 of
                           flip-flop SR_H2.
ST    H2                   Assignment of read value to the
                           variable H2.
```

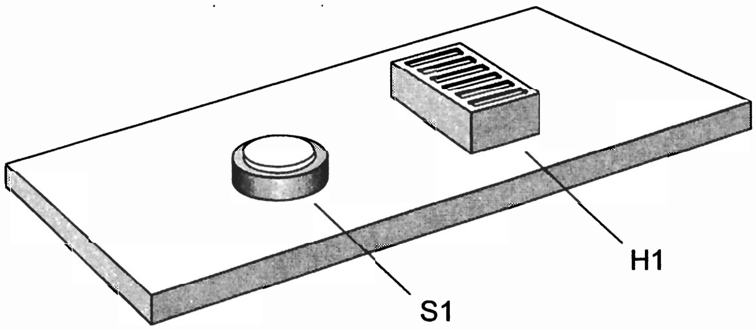
ST

```
SR_H2 (S1  :=  S2);        Invocation of flip-flop SR_H2 using
                           a current transfer parameter.
H2  :=  SR_H2.Q1;          Assignment of output value Q1 of
                           SR_H2 to the variable H2.
```

Problem description

Buzzer H1 is to be switched on by pressing an indicator push button S1.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commissioning of the PLC program and system

Implementation

9.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Signal input, electrical
1	Signalling device

- **Prior to wiring the installation:**
  Switch off power supply!

- Establish the electrical connections.

9.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

9.3 Formulation of PLC program into one of the PLC programming languages

- Select one of the languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

9.4 Testing and commissioning of the PLC program and system

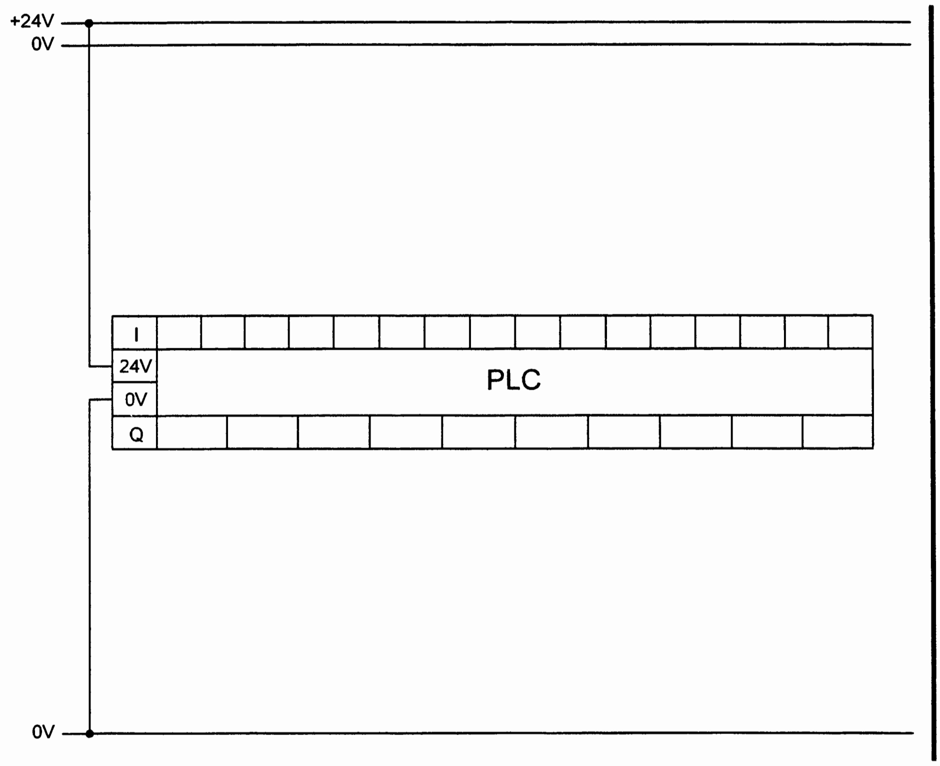
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

9.1 Drawing up the electrical circuit diagram and constructing the circuit

Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

9.2 Declaration of the PLC program variables

Declare the variables required in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

9.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task in one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

WORKSHEET

PLC program

Mark the characteristics applicable to a function or function block.

Set of function and function block

Characteristic	*Function*		*Function block*	
Name	☐ yes	☐ no	☐ yes	☐ no
Input parameters	☐ one	☐ several	☐ one	☐ several
Output parameters	☐ one	☐ several	☐ one	☐ several
Component part of the declaration of variabls	☐ yes	☐ no	☐ yes	☐ no
Status information	☐ yes	☐ no	☐ yes	☐ no

Question:

Which status information is stored in the SR function block?

Subject

Programmable logic controllers

Title

Drill breakage monitoring
Setting and resetting of an output

Training aim

- To be able to set and reset a stored output of a PLC

Technical knowledge

Function block RS, dominant resetting flip-flop

Function block RS contains a dominant resetting flip-flop.

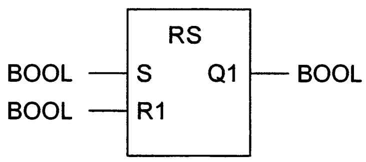


Fig. A10.1: Function block RS

The behaviour of the block represented in Fig. A10.1 is as follows:

- A 1-signal at reset input R1 sets the value of Q1 to 0, irrespective of which value applies at input S.
- A 1-signal at set input S sets output Q1 to 1 only if a 0-signal simultaneously applies at the R1 input.
- If 1-signals apply both at inputs S and R1, output Q1 is reset.

Realisation of the "Reset" storage function in the individual programming languages

The solutions for the example below are listed in table A10.1.

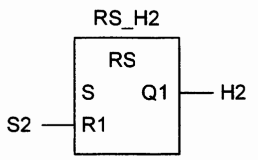
Example

Actuation of push button S2 is to cause lamp H2 to be switched off.

Table A10.1: The storage function reset

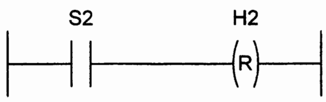
```
VAR
   S2  AT  %IX5   :  BOOL;    (* Switch S2 at input IX5  *)
   H2  AT  %QX6   :  BOOL;    (* Lamp H2 at output QX6   *)
   RS_H2          :  RS;      (* RS flip-flop named RS_H2*)
                              (* For storage of status   *)
                              (* of H2                   *)
END_VAR
```

FBD



RS flip-flop RS_H2 for storage of status of the variable H2. Direct assignment of value of Q1 to the variable H2.

LD



Rung with normally open contact and reset coil.

Table A10.1: The storage function reset (continuation)

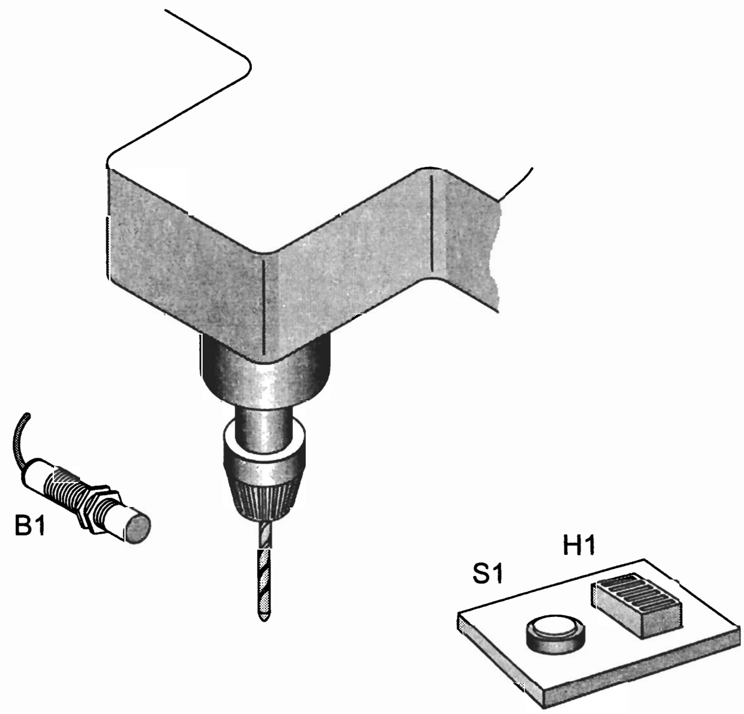
IL	
`LD S2`	Reading of value of S2
`R H2`	Resetting of variable of H2
`or`	
`CAL RS_H2 (R1 := S2)`	Invocation of flip-flop RS_H2 using current transfer parameter S2.
`LD RS_H2.Q1`	Reading of output value Q1 of flip-flop RS_H2.
`ST H2`	Assignment of the read value to the variable H2.
ST	
`RS_H2 (R1 := S2);`	Invocation of flip-flop RS_H2 using a current transfer parameter.
`H2 := RS_H2.Q1;`	Assignment of output value Q1 of RS_H2 to the variable H2.

Problem description

The drill on a drilling unit is monitored by means of a drill breakage sensor (B1).

If the drill is broken, the sensor interrupts the circuit. A buzzer (H1) is to sound in this event. The buzzer can only be switched off via push button S1.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commissioning of the PLC program and system

Implementation

10.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Signal input, electrical
1	Signalling device
1	Proximity switch, optical

- **Prior to wiring the installation:**
  Switch off power supply!

- Establish the electrical connections.

10.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

10.3 Formulation of PLC program into one of the PLC programming languages

- Select one of the languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LDR, FBD, IL and ST.

10.4 Testing and commissioning of the PLC program and system

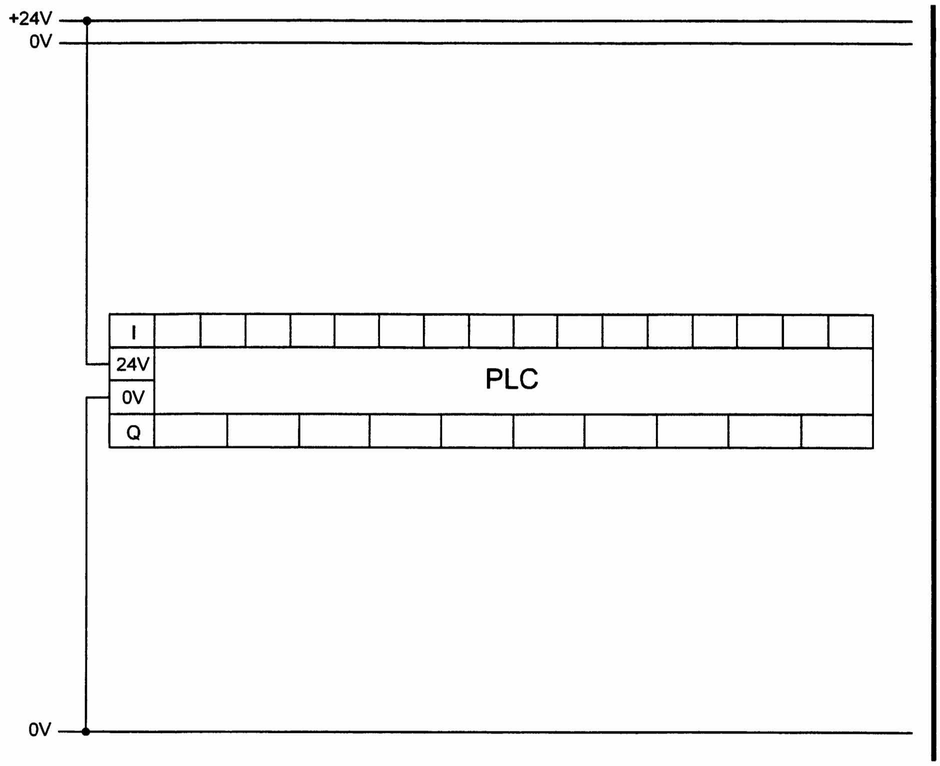
- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

10.1 Drawing up the electrical circuit diagram and constructing the circuit

Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

10.2 Declaration of the PLC program variables

Declare the variables required in the PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

10.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

WORKSHEET

PLC program

Question:

What is the resulting program sequence if a set dominant flip-flop is used instead of a reset dominant flip-flop?

Subject

Programmable logic controllers

Title

Activating a cylinder
Signal edges

Training aim

- To be able to describe the function of signal edges
- To be able to use the standard function block R_TRIG for the recognition of a rising signal edge

Technical knowledge

PLC applications frequently require the detection and evaluation not of a signal itself, but of the point of change of a signal. These signal changes are described as edges.

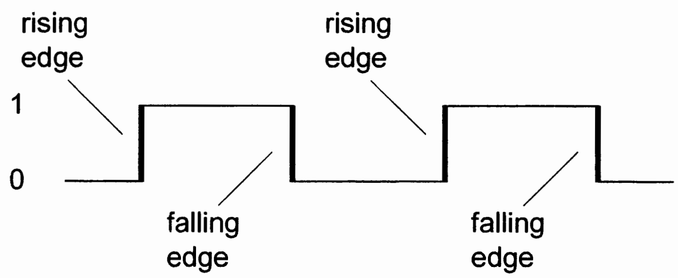


Fig. A11.1: Edges

Rising (positive) edges mark the instant, during which a signal change takes place from 0 to 1.

Falling (negative) edges mark the instant, during which a signal change takes place from 1 to 0.

Function block R_TRIG for rising edge detection.This standard function block is used for the detection of a rising edge.

R_TRIG

BOOL — CLK Q — BOOL

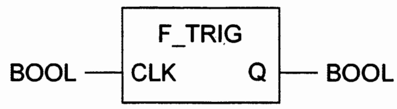
Fig. A11.2: Function block R_TRIG

If a signal change takes place from 0 to 1 at input CLK, output Q assumes the value 1 during a program cycle.

Function block F_TRIG for falling edge detection

This standard function block is used for the detection of a falling edge.

Fig. A11.3:
Function block F_TRIG



If a signal change takes place from 1 to 0 at input CLK, output Q carries a 1-signal during a program cycle.

Realisation of edge evaluation in the individual programming languages

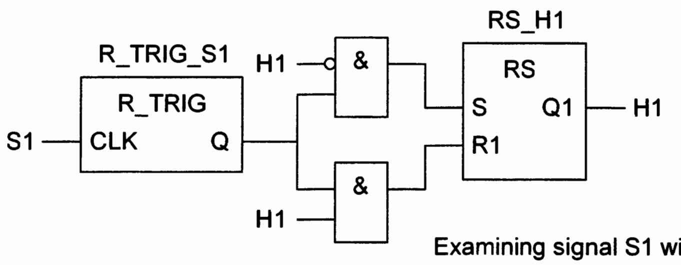
Edge evaluation has been programmed in the languages FBD, LD, IL and ST for the example below.

Example Actuation of a push button S1 causes a lamp H1 to be switched on. Repeat actuation of push button S1 switches off the lamp again.

Table A11.1:
Evaluation of a rising edge

```
VAR
   S1  AT    %IX1   :  BOOL;    (* Push button S1         *)
   H1  AT    %QX1   :  BOOL;    (* Lamp H1                *)
   R_TRIG_S1        :  R_TRIG   (* Rising edge of S1      *)
   RS_H1            :  RS;      (* Flip-flop for H1       *)
END_VAR
```

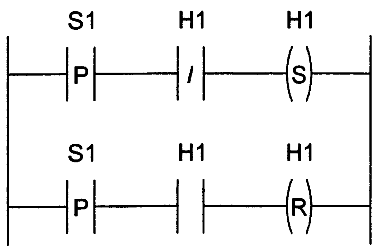
FBD



Examining signal S1 with function block R_TRIG_S1 for rising edge. Depending on the status of lamp H1, a positive results leads to H1 being switched on or off.

Table A11.1: (Continuation)

LD



Detection of a positive edge by means of a special edge contact. Depending on the status of the lamp, this may be switched on in the first rung and switched off in the second rung.

IL

```
CAL     R_TRIG_S1 (CLK := S1)    Invocation of function
                                 block R_TRIG_S1.

LD      R_TRIG_S1.Q              Depending on the result of
ANDN    H1                       the edge evaluation, the
S       H1                       lamp switched on ...

LD      R_TRIG_S1.Q
AND     H1
R       H1                       or off.
```

ST

```
R_TRIG_S1 (CLK := S1);           Invocation of function block
                                 R_TRIG_S1.

RS_H1(S:= R_TRIG_S1.Q & NOT H1,  Invocation of flip-flop
      R1:= R_TRIG_S1.Q & H1);    RS_H1.

H1 := RS_H1.Q1;                  Status of flip-flop RS_H1
                                 is mapped to H1.
```

Comment

Since the conditions for switching the lamp on and off are mutually exclusive, a set dominant flip-flop may also be used instead of a reset dominant one.

Realisation of edge evaluation without special edge function blocks

If a PLC system does not support special function blocks for the detection of edges, memories may be used for the detection of signal changes.

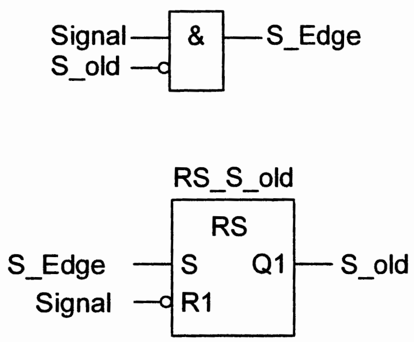
PLC programs are continually cyclically processed. In order to detect a signal change, it is necessary to check whether the status of a signal has changed from one processing cycle to the next. To do this, the old signal status has to be stored and compared with the new current status.

Fig. A11.4 illustrates the method used to detect a rising edge.

Fig. A11.4: Detecting a rising edge

```
VAR
  Signal    AT %IX1  :  BOOL;       (* current input signal   *)
  S_Edge    AT %MX1  :  BOOL;       (* detects edge of the    *)
                                    (* input signal           *)
  S_old     AT %MX2  :  BOOL := 0;  (* stores old status of   *)
                                    (* input signal           *)
  RS_S_old           :  RS;         (* flip-flop for memory   *)
                                    (* S_old                  *)

END_VAR
```



The signal to be examined is represented by the variable "Signal". The old status of the variable "Signal" is stored in the memory "S_Old". If a rising edge occurs, the memory "S_Edge" assumes the value 1 for one processing cycle.

It should be noted that the memory "S_Old" must maintain the value 0 at the program start (in the machining cycle).

The program parts shown have been formulated in the language FBD as an example.

The evaluation of a falling edge may be realised as illustrated below.

```
VAR
  Signal    AT %IX1   :  BOOL;        (* current input signal    *)
  S_Edge    AT %MX1   :  BOOL;        (* detects edge of the     *)
                                      (* input signal            *)
  S_old     AT %MX2   :  BOOL := 0; (* stores old status of    *)
                                      (* input signal            *)
  SR_S_old            :  SR;          (* flip-flop for memory    *)
                                      (* S_old                   *)

END_VAR
```

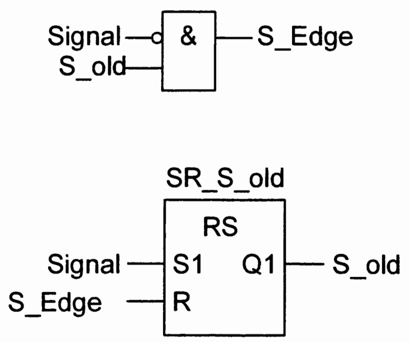
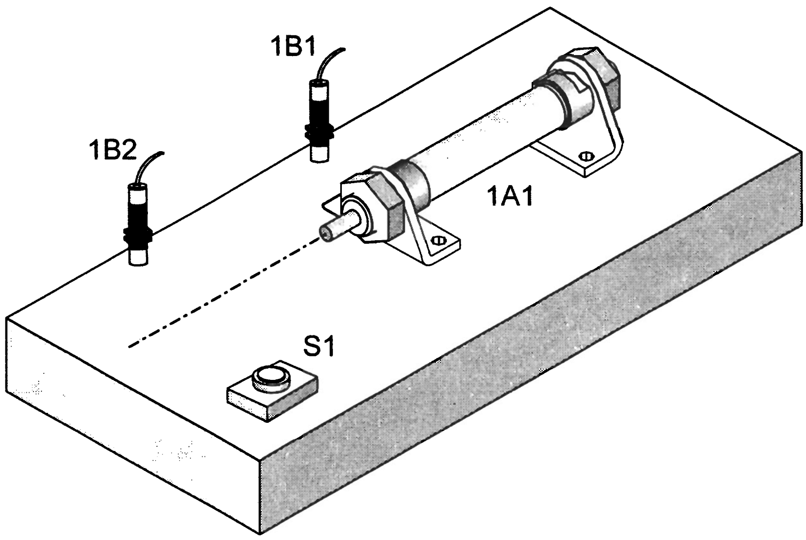


Fig. A11.5:
Detecting of a falling edge

Problem description

A cylinder is actuated by means of a spring-returned solenoid valve (coil 1Y1). Two proximity switches signal the positions "extended" (1B2) and "retracted" (1B1). Push button (S1) is used to actuate the cylinder in such a way that it advances from the retracted end position into the opposite direction. The cylinder must advance only once per push button actuation. To trigger a second movement of the cylinder, the push button must be released and actuated afresh.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commissioning of the PLC program and system

11.1 Drawing up the electrical circuit diagram and constructing the circuit

Implementation

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
1	5/2-way single solenoid valve
1	Signal input, electrical
1	Proximity switch, inductive
1	Proximity switch, capacitive
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air supply at service unit!

- Establish the electrical and pneumatic connections.

11.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section are represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

11.3 Formulation of the PLC program into one of the PLC programming languages

- Select one of the languages supported by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

11.4 Testing and commissioning of the PLC program and system

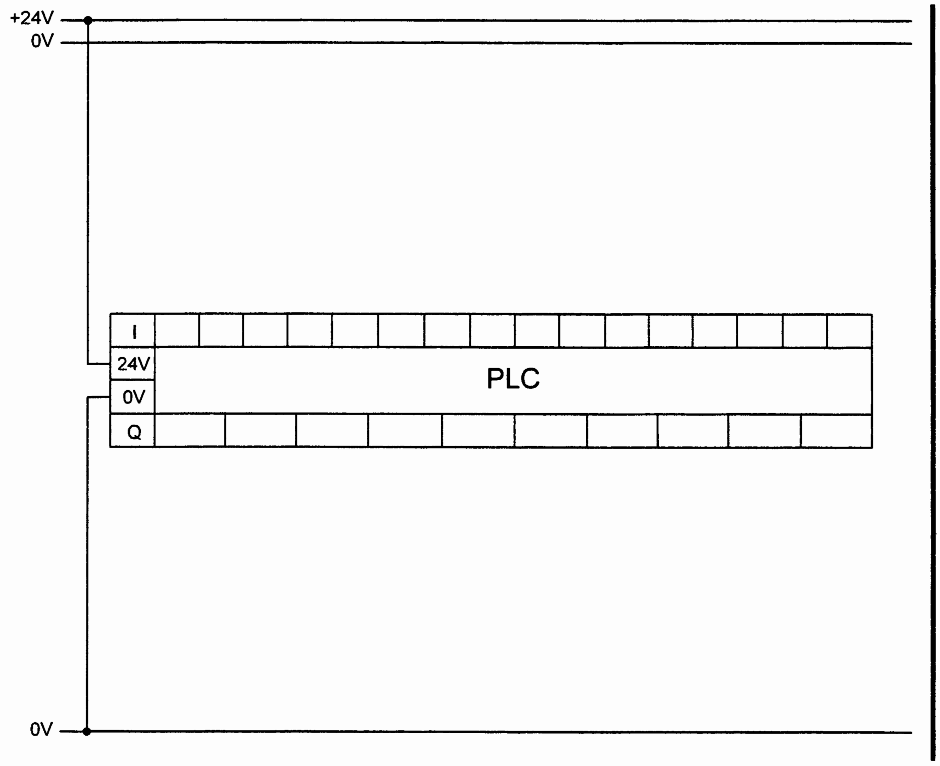
- **Prior to commissioning of the installation:**
  Check assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply to operating pressure (see data sheets for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

11.1 Drawing up the electrical circuit diagram and constructing the circuit

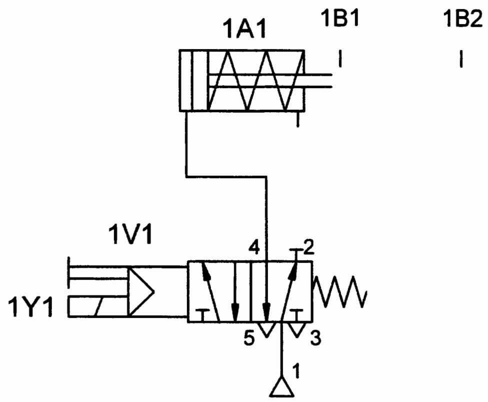
Complete the electrical circuit diagram and enter the input and output addressed available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



11.2 Declaration of the PLC program variables

Declare the variables required in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

WORKSHEET

11.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Question 1:

What is understood by a negative edge?

Question 2:

What effect does the period of actuation have on the program execution?

Programmable logic controllers — *Subject*

Bonding of components
Pulse — *Title*

- To be able to use standard function block TP for pulse time response — *Training aim*

Technical knowledge

A large number of control tasks require the programming of time. Through EN 61131-3 (IEC 61131-3), standard function blocks are available for timers with different time response.

Timers are available for the realisation of a pulse time response, a switch-on signal delay and a switch-off signal delay.

Function block TP, pulse timer

Standard functon block TP (Fig. A12.1) is a pulse timer

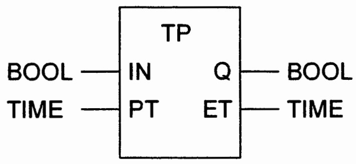


Fig. A12.1:
Function block TP

The response of function block TP is as follows:

- Function block TP is started via a short or long signal at input IN.
- Once the timer has started, a 1-signal applies at output Q for the time specified at input PT.
- The current timer value (the time, which has elapsed since the start) is available at the output ET.
- The timer can only be started again once it has expired.

Programming of a pulse timer in the individual languages

The use of a pulse timer in the individual languages is illustrated with the help of the example given below.

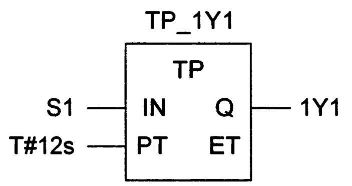
Example

Workpieces are clamped securely for a period of 12 seconds for a machining process by means of a special device on a cylinder 1Y1. The process is triggered by acuating a start button S1.

Table A12.1: Use of a pulse timer

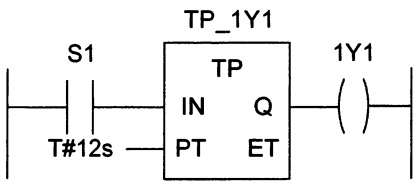
```
VAR
  S1   AT  %IX1   :  BOOL;   (* Push button S1          *)
  1Y1  AT  %QX1   :  BOOL;   (* Coil 1Y1 for cylinder   *)
  TP_1Y1          :  TP;     (* Pulse timer named       *)
                             (* TP_1Y1 for clamping process *)
END_VAR
```

FBD



Timer function block TP_1Y1, connected with the current parameters.

LD



Interconnecting timer function block TP_1Y1 into the rung.

Table A12.1: Use of a pulse timer continuation

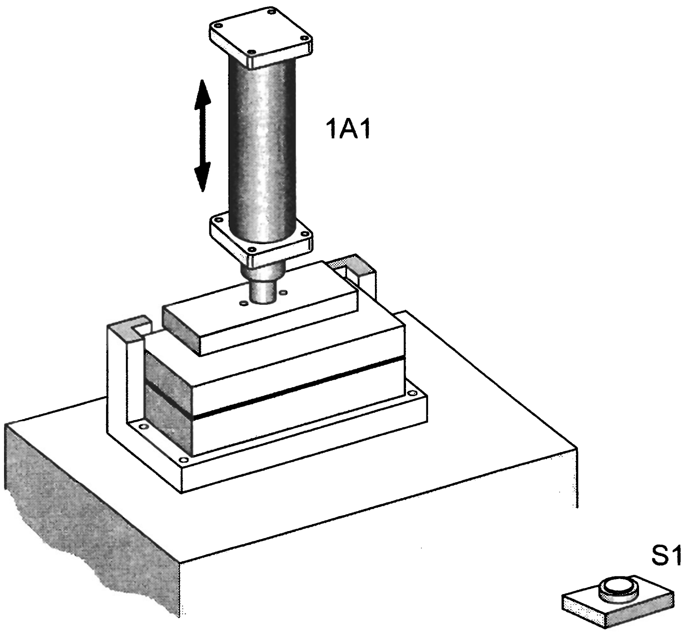
IL	
CAL TP_1Y1 (IN := S1, PT := T#12s)	Invocation of function block TP_1Y1.
LD TP_1Y1.Q	Reading of output Q of TP_1Y1.
ST 1Y1	Storage of current result to Y1.

ST	
TP_1Y1 (IN := S1, PT := T#12s);	Invocation of function block TP_1Y1.
1Y1 := TP_1Y1.Q;	Assignment of output Q of TP_1Y1 to 1Y1.

Problem description

Two components are to be bonded together with the help of a pneumatic cylinder 1A1. To do this, the bonding surfaces are pressed together with a defined force for 5 seconds. The time is commenced once the cylinder advanves from its reacted end position (sensor 1B1=1). Once the 5 seconds have expired, the cylinder is to return to the initial position. The bonding process is started by a push button S1.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commisioning of the PLC program system

12.1 Drawing up the electrical circuit diagram and constructing the circuit

Implementation

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
1	5/2-way single solenoid valve
1	Signal input, electrical
1	Proximity switch, inductive-magnetisch
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air suplp at service unit!

- Establish the electrical and pneumatic connections.

12.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address - only if directly addressed variables are used - and variables comments.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

12.3 Formulation of the PLC program into one of the PLC programming languages

- Select one of the languages supoprted by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

12.4 Testing and commisioning of the PLC program system

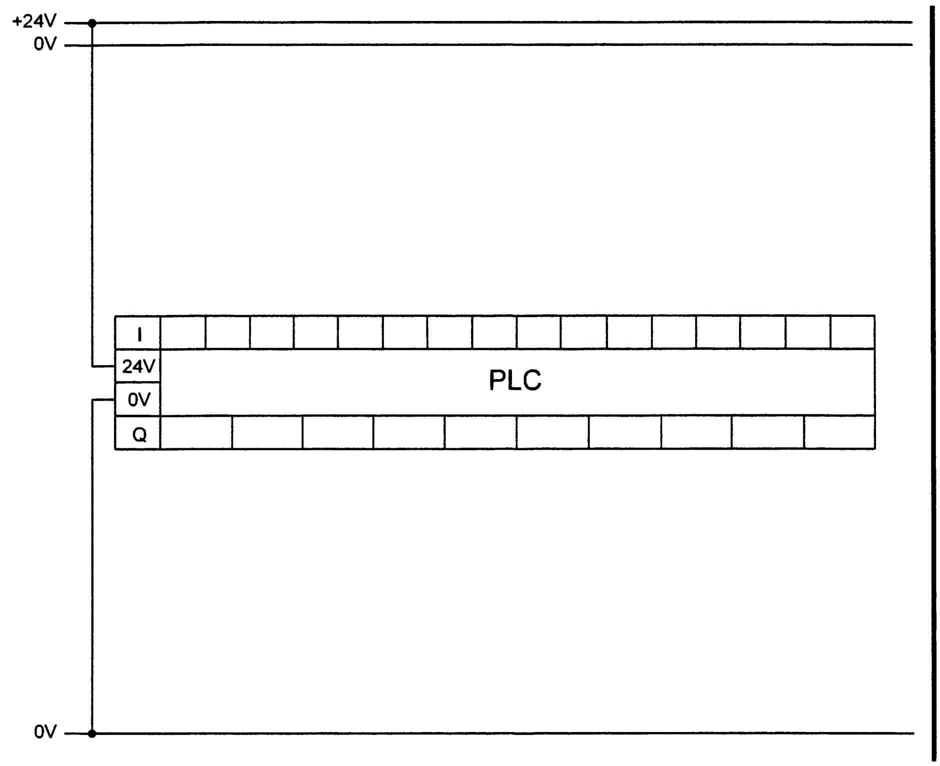
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

12.1 Drawing up the electrical circuit diagram and constructing the circuit

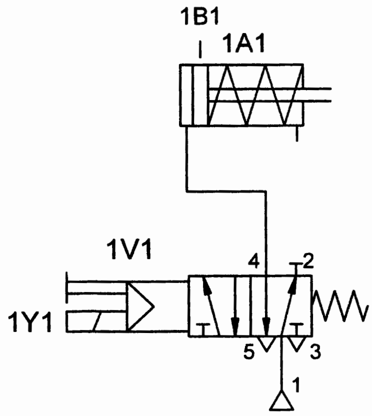
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



12.2 Declaration of the PLC program variables

Declare the variables reqired in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

WORKSHEET

12.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Question 1:

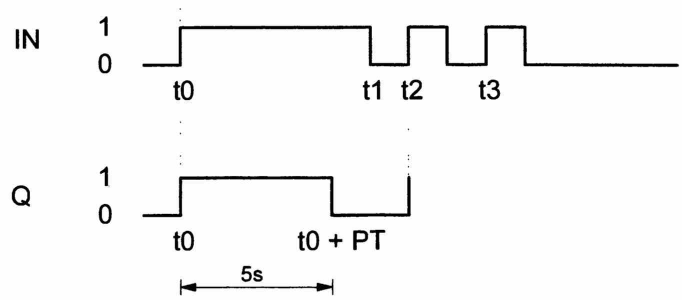
Specify the name and the function of the parameters of the pulse timer.

Question 2:

What is the response of the timer, if a new start signal is given prior to the timer expiring?

Complete the diagram.

Time response of Pulse timer



Subject: *Programmable logic controllers*

Title:

Embossing device
Switch-on signal delay

Training aim

- To be able to realise a switch-on signal delay by using the standard function block TON

Technical knowledge

Function block TON, Switch-on signal delay

The standard function block TON is used to generate a switch-on signal delay.

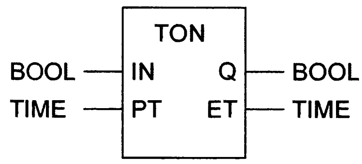


Fig. A13.1:
Function block TON

The behaviour of function block TON is as follows:

- Function block TON is started by means of a 1-signal at input IN.
- Upon expiry of the time specified at input PT, output Q carries a 1-signal. The 1-signal at output Q applies until the input signal IN reverts to the value 0.
- If the duration of the input signal IN is shorter than the specified time PT, the value of output Q remains a constant 0.
- The current timer value (the time, which has elapsed since the start) is available at output ET.

Programming of a switch-on signal delay in the individual languages

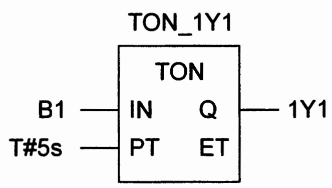
The use of a switch-on signal delay is demonstrated in the following example:

Example The door of a bus will only close when the boarding area has been clear for a specified period (5 seconds). This is monitored by means of a light barrier.

Table A13.1: Use of a Switch-on signal delay

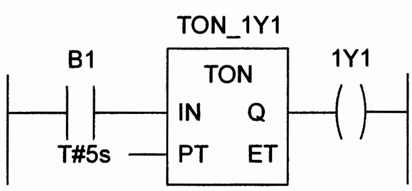
```
VAR
    B1    AT  %IX1  :BOOL;   (* Light barrier                 *)
    1Y1   AT  %QX1  :BOOL;   (* Coil 1Y1 for cylinder for closing *)
                             (* the door                      *)
    TON_1Y1         :TON;    (* Switch-on signal delay named  *)
                             (* TON_1Y1  for closing of door  *)
END_VAR
```

FBD



Connecting the inputs and outputs of function block TON_1Y1 with current parameters.

LD



Interconnection of function block TON_1Y1 into the rung.

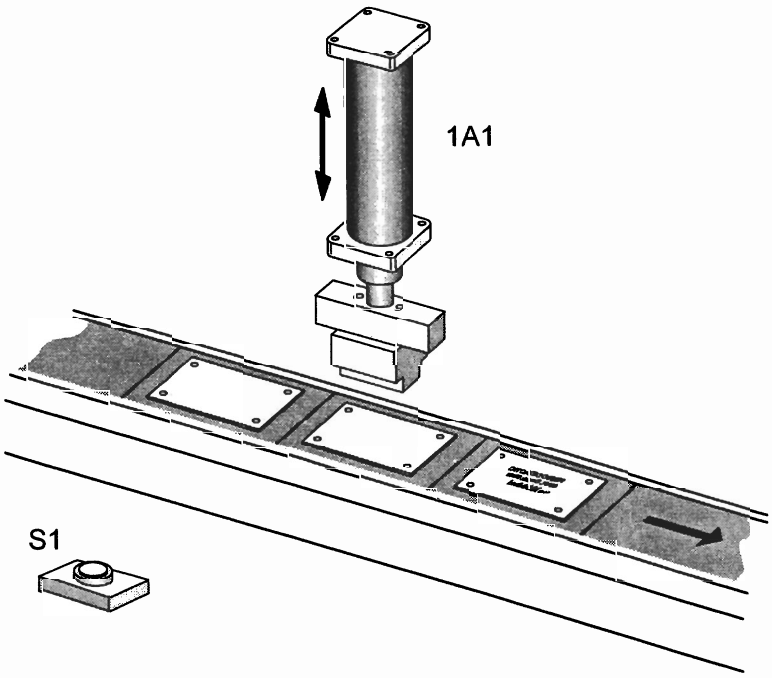
Table A13.1: Use of a Switch-on signal delay (continuation)

IL	
CAL TON_1Y1 (IN := B1,PT := T#5s)	Invocation of function block TON_1Y1.
LD TON_1Y1.Q	Reading of output Q of TON_1Y1.
ST 1Y1	Storing of current result to 1Y1.
ST	
TON_1Y1 (IN := B1, PT := T#5s);	Invocation of function block TON_1Y1.
1Y1 := TON_1Y1.Q;	Assignment of output Q of TON_1Y1 to 1Y1.

Problem description

A workpiece is to be embossed by activating a start button (S1). In order to ensure that the embossing cycle is not triggered inadvertently, the embossing cycle is to be triggered only after 3 seconds have expired. During this time the start button must be permanently actuated. The position of the cylinder 1A1 is established by means of the proximity switches 1B1 (retracted) and 1B2 (extended).

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commisioning of the PLC program system

13.1 Drawing up the electrical circuit diagram and constructing the circuit

Implementation

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
1	5/2-way single solenoid valve
1	Signal input, electrical
1	Proximity switch, inductive
1	Proximity switch, capacitive
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air supply at service unit!

- Establish the electrical and pneumatic connections.

13.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address - only if directly addressed variables are used - and variables comments.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

13.3 Formulation of the PLC program into one of the PLC programming languages

- Select one of the languages supoprted by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

13.4 Testing and commisioning of the PLC program system

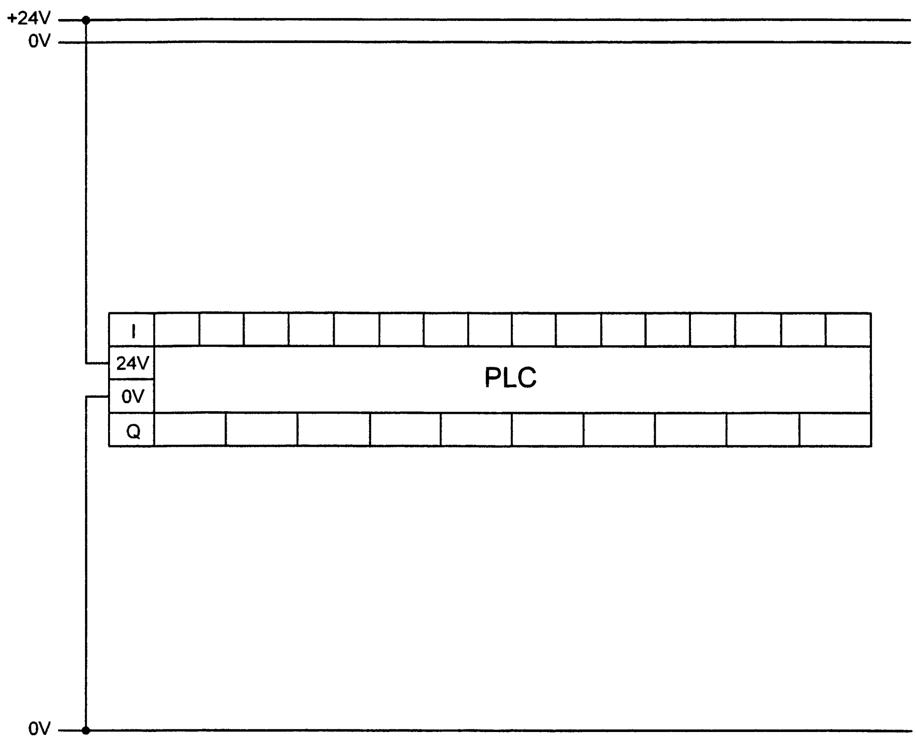
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

13.1 Drawing up the electrical circuit diagram and constructing the circuit

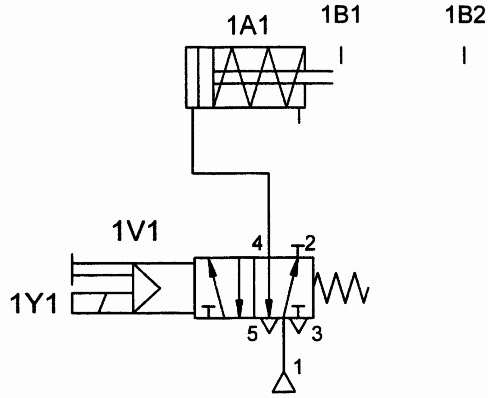
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



13.2 Declaration of the PLC program variables

Declare the variables reqired in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

WORKSHEET

13.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Question:

The embossing cycle has been initiated. However, the start button is released before the 3 seconds have expired. What effect does this have on the program execution?

Subject

Programmable logic controllers

Title

Clamping device
Switch-off signal delay

Training aim

- To be able to realise a switch-off signal delay by using the standard function block TOF

Technical knowledge

Function block TOF, switch-off signal delay

The standard function block TOF (Fig. A14.1) is used to generate a switch-off signal delay.

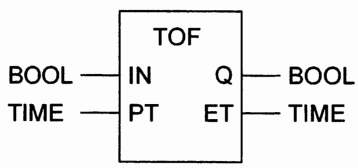


Fig. A14.1:
Function block TOF

The behaviour response of function block TOF is as follows:

- Function block TOF is started via a 1-signal at input IN. Output Q simultaneously receives the value 1.
- After the input signal IN has reverted to the value 0, the 1 signal continues to be applied at output Q for the time specified at the PT input and then returns to the value 0.

Programming of a switch-off signal delay in the individual languages

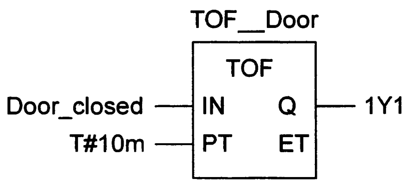
The use of a switch-off signal delay is demonstrated in the following:

Example The door of a furnace includes a lock so that it cannot be opened instantly during the burning process. If a signal is given to open the door, this will only be unlocked after 10 minutes has expired.

Table A14.1: Use of a switch-off signal delay

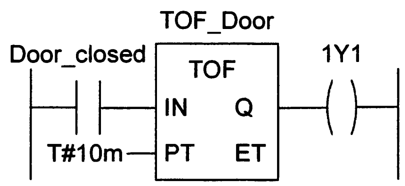
```
VAR
   Door_closed  AT %MX1 : BOOL;   (* Memory for latching of   *)
                                  (* furnace door             *)
      1Y1       AT %QX1 : BOOL;   (* Coil 1Y1 for cylinder    *)
                                  (* opening of furnace door  *)
   TOF_Door             : TOF;    (* Switch-off signal delay  *)
                                  (* named TOF_Door           *)
END_VAR
```

FBD



Connection of inputs and outputs of function block TOF_Door with current parameters.

LD



Interconnection of function block TOF_Door into the rung.

Table A14.1:
Use of a switch-off signal delay (continuation)

IL

```
CAL TOF_Door (IN := Door_closed, PT := T#10m)Invocation of
                                             function block
LD   TOF_Door.Q                              TOF_Door.
                                             Reading of output
ST   1Y1                                     Q of TOF_Door.
                                             Storage of current
                                             result to 1Y1.
```

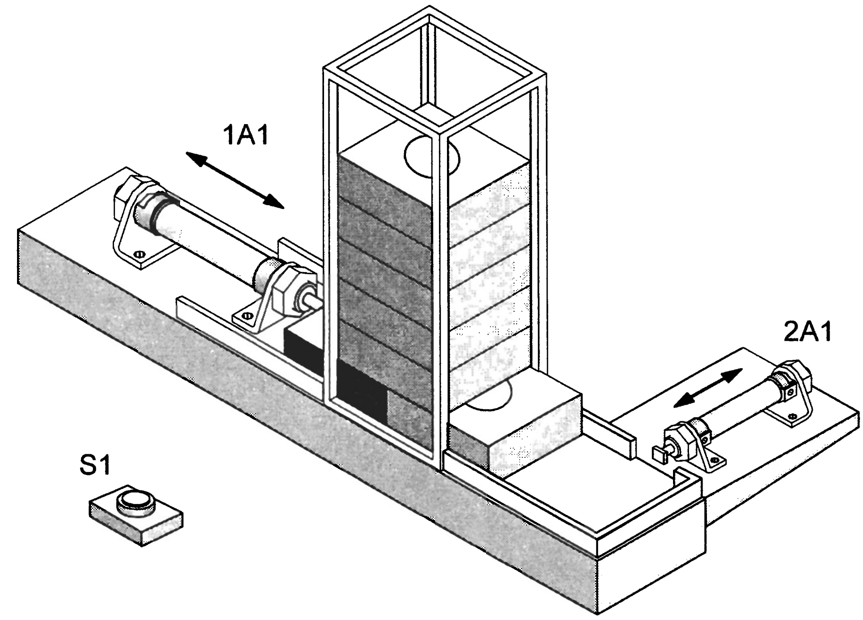
ST

```
TOF_Door (IN := Door_closed, PT := T#10m); Invocation of
                                           functionblock
                                           TOF_Door.
1Y1  :=  TOF_Door.Q;                       Assignment of output
                                           Q of TOF_Door to 1Y1.
```

Problem description

A workpiece is to be clamped by activating the start button S1. When the workpiece is clamped by cylinder 1A1, cylinder 2A1 extends and embosses the workpiece. Since the workpiece requires time to cool down, it remains clamped for a period of 3 seconds. This time is started with the advancing of cylinder 1A1.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Declaration of the PLC program variables
3. Formulation of the PLC program into one of the PLC programming languages
4. Testing and commisioning of the PLC program system

14.1 Drawing up the electrical circuit diagram and constructing the circuit

Implementation

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
1	Double-acting cylinder
1	5/2-way single solenoid valve
1	5/2-way double solenoid valve
1	Signal input, electrical
4	Proximity switch, inductive
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air suplp at service unit!

⚠

- Establish the electrical and pneumatic connections.

14.2 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

14.3 Formulation of the PLC program into one of the PLC programming languages

- Select one of the languages supoprted by your PLC system for programming. Suitable languages for the formulation of logic control systems are LD, FBD, IL and ST.

14.4 Testing and commisioning of the PLC program system

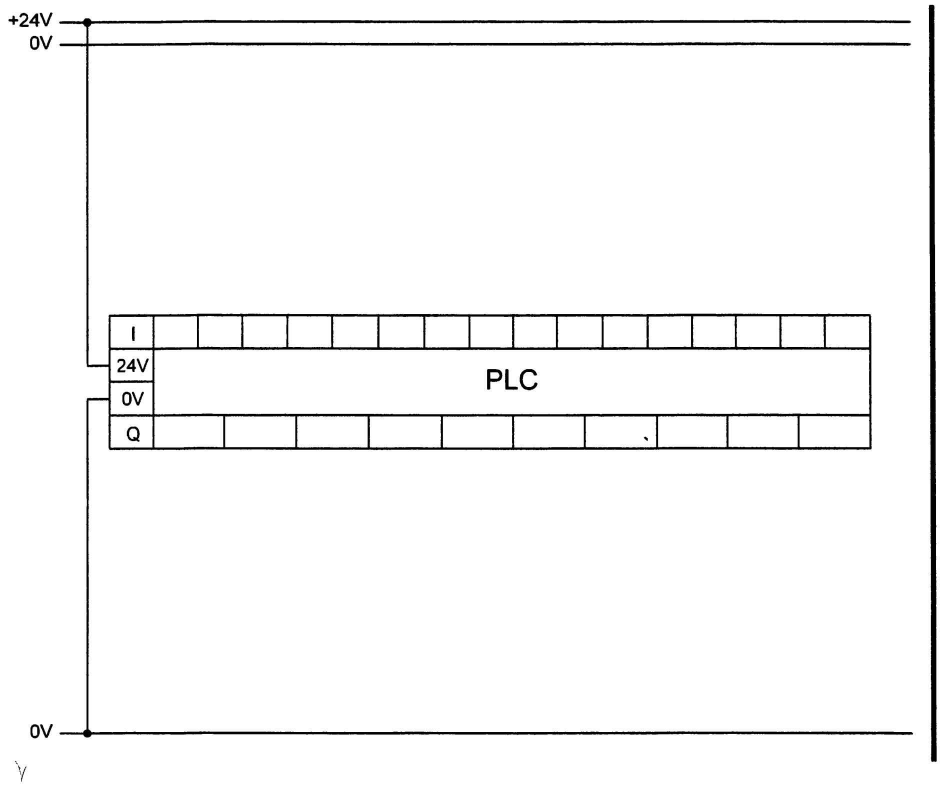
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

14.1 Drawing up the electrical circuit diagram and constructing the circuit

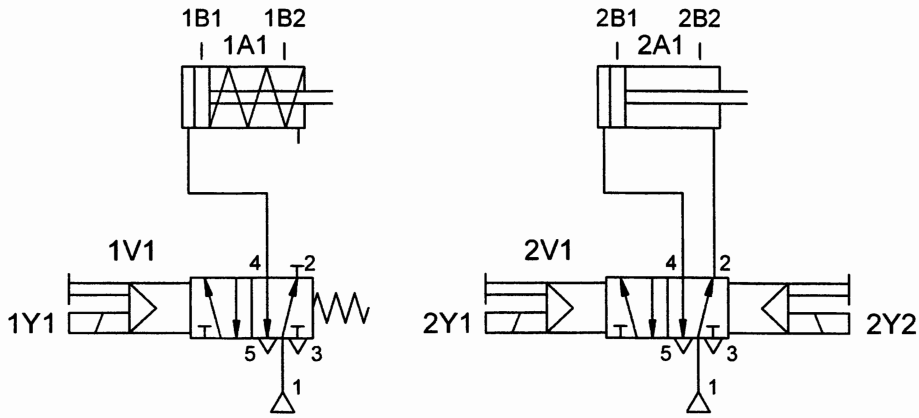
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



WORKSHEET

14.2 Declaration of the PLC program variables

Declare the variables reqired in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

14.3 Formulation of the PLC program into one of the PLC programming languages

Formulate the solution of the control task into one of these languages:

- Function block diagram (FBD)
- Ladder diagram (LD)
- Instruction list (IL)
- Structured text (ST)

PLC program

Question:

Through which signal is the time for the switch-off signal delay started?

Subject

Programmable logic controllers

Title

Lifting device for packages
Linear sequence

Training aim

- To be able to design and represent simple sequence control systems in accordance with IEC 60848.
- To be able to program a sequence control system consisting of a linear sequence
- To be able to use the programming language Sequential Function Chart

Technical knowledge

Sequence control systems

Sequence control systems describe processes, which proceed in several clearly separated steps.

The transition from one step to the next is dependent on the process statuses. It is possible for the process to be branched into partial processes in relation to the process statuses established.

The program of a sequence control system must therefore fulfil three basic exercises:

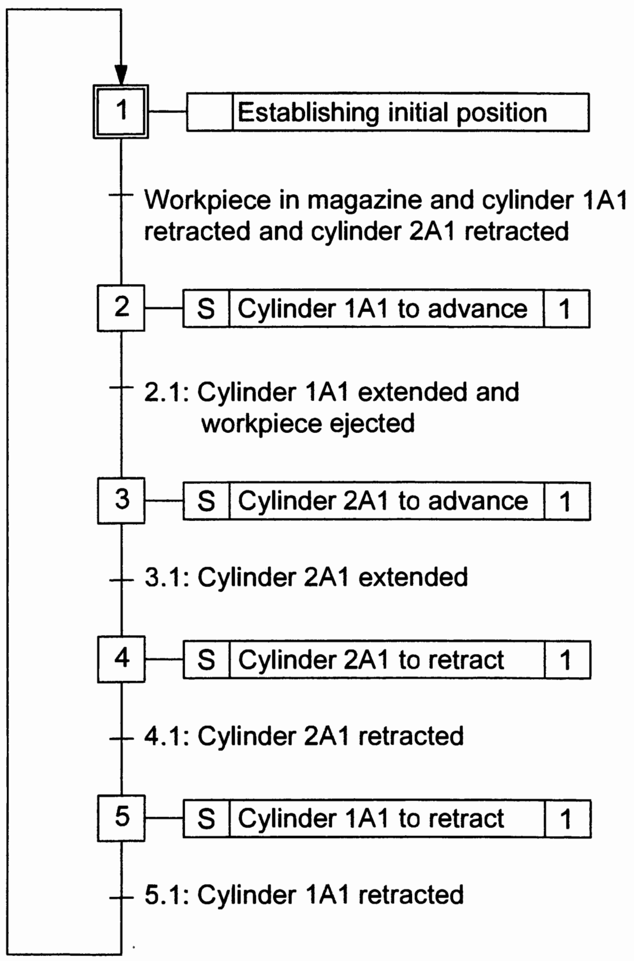
Executing actions: Step	• Setting and resetting of outputs • Setting and resetting of memories • Setting and starting timers and counters
Interrogating events: Transition conditions (Step enabling condition)	• Interrogating inputs and memories • Interrogating timers • Interrogating counter status
Establishing paths for program continuation: Branching	• Selecting path • Following parallel path

Fig. A15.1: Fuctions of a control program

General representation of a sequence control system

Function chart in accordance with IEC 60848 is for the description and planning of a sequence control system. This permits a clear, graphic representation of the behaviour and function of a sequence control system.

Fig. A15.2: Example of a control program



The example shown above describes the following control task:

A workpiece is made available from a magazine for further machining. The workpiece is ejected from the magazine by a cylinder 1A1, and then transferred onto a conveyor belt via a slide by means of a second cylinder 2A1.

Programming of a sequence control system in Sequential Function Chart

Sequence control systems can be easily and clearly programmed in a sequential function chart. The sequential function chart is derived from the function chart in accordance with IEC 60848.

The example shown Fig. A15.3 illustrates the use of sequential function hart for the control task mentioned above.

- Step => Classification into actions
- Transition => Description by means of transition condition
- Alternative branch and junction
- Parallel branch and junction

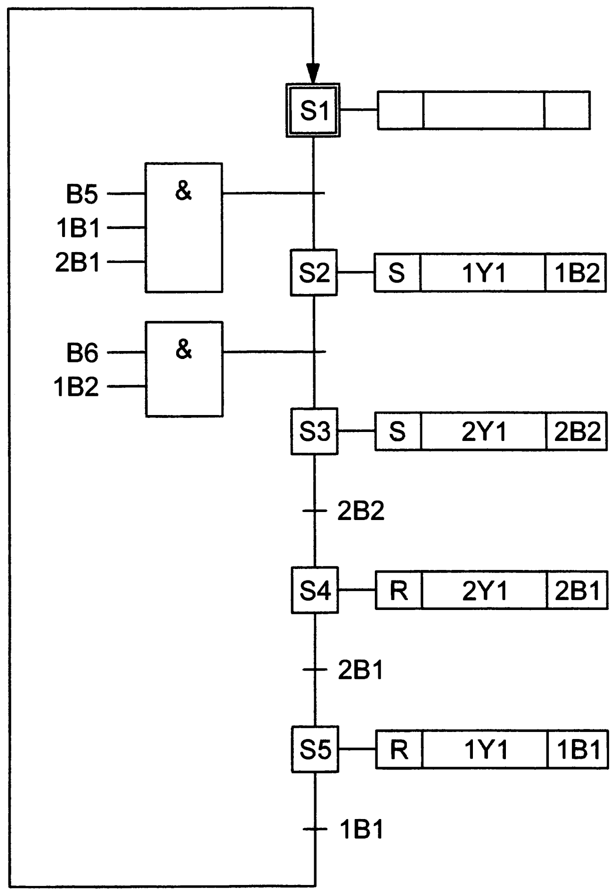
When the PLC program is started, the step designated as initial step S1 automatically becomes active. A system is frequently moved into the initial position as a result of the initial step. In the example shown, step S1 is a void step. If the subsequent step enabling conditions – cylinder 1A1 and 2A1 are retracted and the magazine contains workpieces – are met, step S2 is set and step S1 reset. It should be noted that the step names represent names in the sense of EN 61131-3 (IEC 61131-3). They must therefore start with a letter or an underline. In addition, insofar as this is possible for the process concerned, a feedback variable is specified in the third field of each action indicating the end of the action.

In step S2, cylinder 1A1 is extended by setting coil 1Y1. When this cylinder has reached its forward end position and the workpiece is in the correct position (B6 = 1), step S2 is reset and step S3 activated. Cylinder 1A1 remains extended as a result of the S-qualifier. In step S3, cylinder 2A1 advances due to coil 2Y1 being set and transfers the workpiece to a slide. Cylinder 2A1 retracts again once it has reached its forward end position. If sensor 2B1 signals that the retracted end position of cylinder 2A1 has been reached, cylinder 1A1 also retracts. Sensor 1B1 now signals the end of the sequence and the program returns to the start. The complete step sequence is repeated again.

Fig. A15.3: Example of sequential function chart

```
VAR
   1Y1  AT  %QX1   :  BOOL;     (* Coil 1Y1 for cylinder 1A1   *)
   2Y1  AT  %QX2   :  BOOL;     (* Coil 2Y1 for cylinder 2A1   *)
   B5   AT  %IX1   :  BOOL;     (* Workpiece in magazine       *)
   B6   AT  %IX2   :  BOOL;     (* Workpiece ejected           *)
   1B1  AT  %IX3   :  BOOL;     (* Cylinder 1A1 retracted      *)
   1B2  AT  %IX4   :  BOOL;     (* Cylinder 1A1 extended       *)
   2B1  AT  %IX5   :  BOOL;     (* Cylinder 2A1 retracted      *)
   2B2  AT  %IX6   :  BOOL;     (* Cylinder 2A1 extended       *)

END_VAR
```



Generating a step sequence by means of RS storage elements

The step sequence may be generated by using storage elements if the programming language Sequential Function Chart is not supported by a PLC program.

Each step is assigned an RS flip-flop. This stores the status of the step. The relevant flip-flop is set, if the step is in the process of being executed; if the step is inactive, the flip-flop is reset.

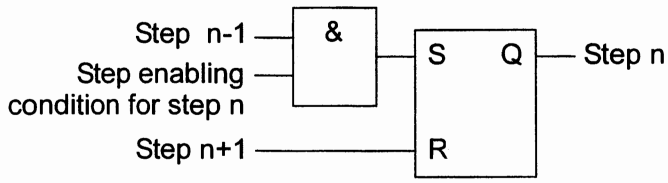


Fig. A15.4:
Representation of a step

As shown in Fig. A15.4, the start conditions for any step n (within a step sequence) are:

- The preceding step n-1 is set
- The step enabling condition for step n is met

Each step is reset by the set subsequent step.

In this way, the individual steps of a step sequence are processed consecutively.

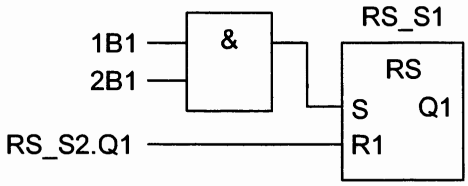
The structure of a step sequence is set out in detail in Fig. A15.5. The language FBD is used for the programming of the control task in Fig. A15.2. All actions occur as boolean actions.

*Fig. A15.5:
Example of a step
sequence with
RS storage elements*

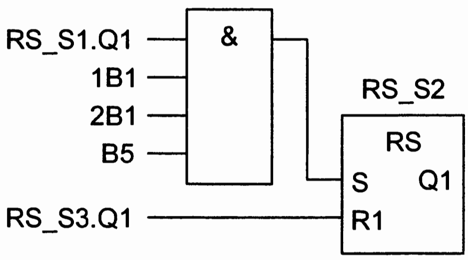
```
VAR
   1Y1  AT  %QX1   :  BOOL;    (* Coil 1Y1 for cylinder 1A1  *)
   2Y1  AT  %QX2   :  BOOL;    (* Coil 2Y1 for cylinder 2A1  *)
   B5   AT  %IX1   :  BOOL;    (* Workpiece in magazine      *)
   B6   AT  %IX2   :  BOOL;    (* Workpiece ejected          *)
   1B1  AT  %IX3   :  BOOL;    (* Cylinder 1A1 retracted     *)
   1B2  AT  %IX4   :  BOOL;    (* Cylinder 1A1 extended      *)
   2B1  AT  %IX5   :  BOOL;    (* Cylinder 2A1 retracted     *)
   2B2  AT  %IX6   :  BOOL;    (* Cylinder 2A1 extended      *)
   RS_S1           :  RS;      (* Flip-flop for Step S1      *)
   RS_S2           :  RS;      (* Flip-flop for Step S2      *)
   RS_S3           :  RS;      (* Flip-flop for Step S3      *)
   RS_S4           :  RS;      (* Flip-flop for Step S4      *)
   RS_S5           :  RS;      (* Flip-flop for Step S5      *)
   RS_1Y1          :  RS;      (* Flip-flop for Coil 1Y1     *)
   RS_2Y1          :  RS;      (* Flip-flop for Coil 2Y1     *)

END_VAR
```

(* Programming of step sequence *)



(* Setting of *)
(* flip-flop for *)
(* Step S1 *)



(* Setting of *)
(* flip-flop for *)
(* Step S2 *)

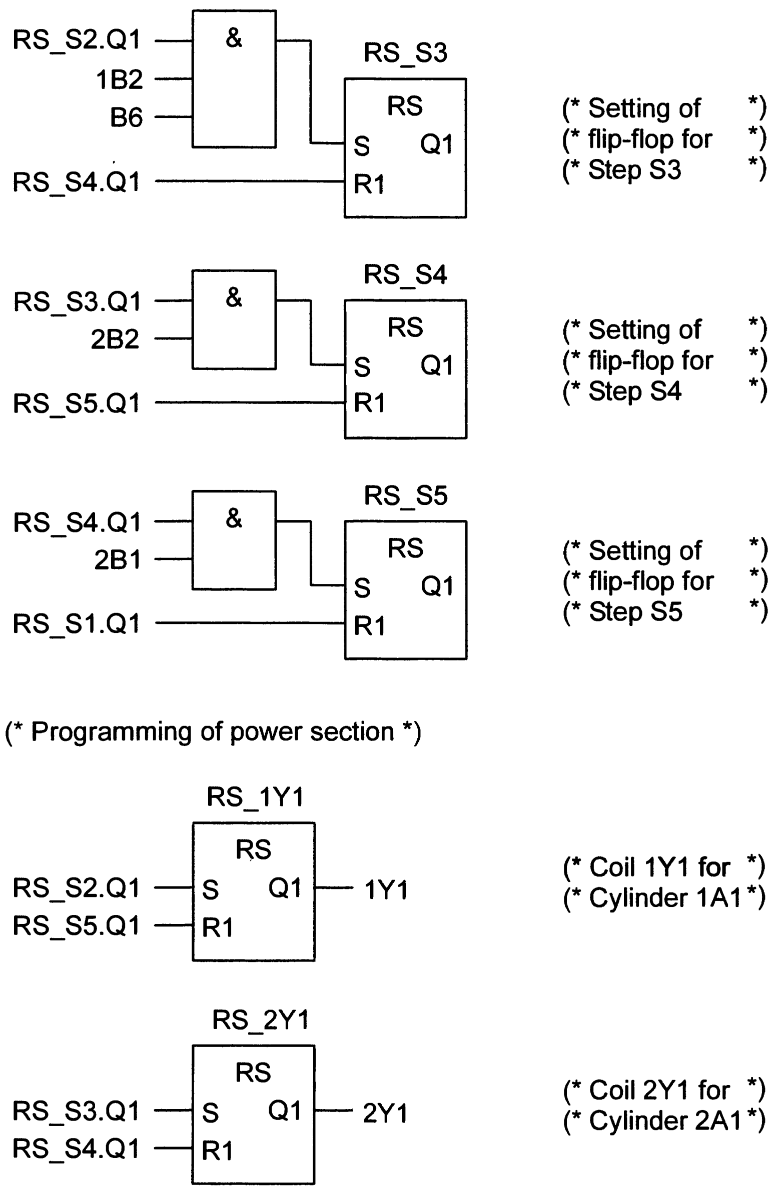


Fig. A15.5: Example of a step sequence with RS storage elements (continuation)

The programming of the step sequence requires an extension of the declaration section in Fig. A15.3.

An RS flip-flop is additionally required for each step. Moreover, the statuses of coils 1Y1 and 2Y1 are stored by means of flip-flops.

The program consists of

- Step sequence
- Power section (for activation of outputs)

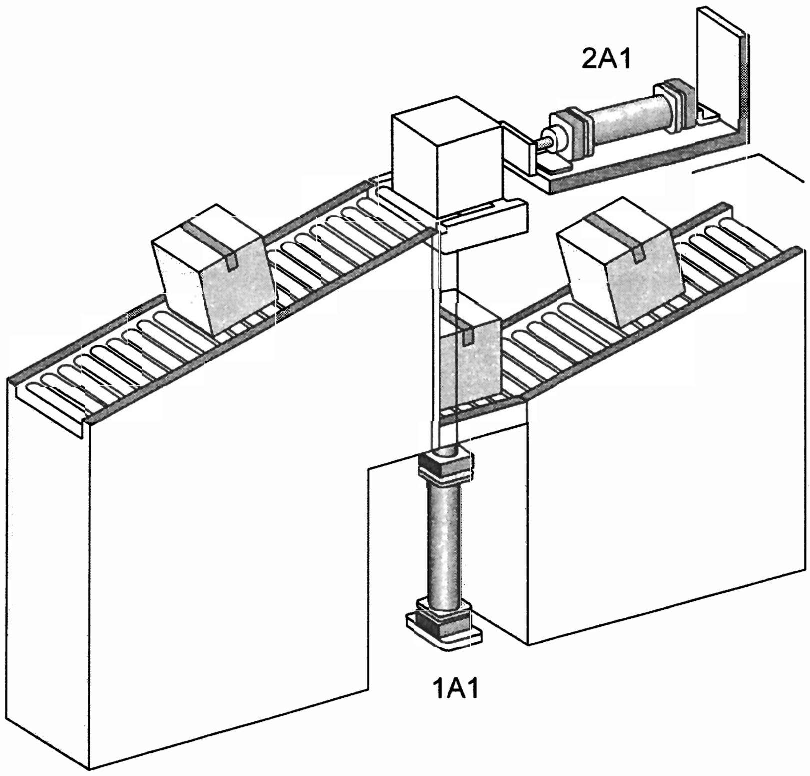
Problem description

A roller conveyor is monitored by a proximity switch B5 as to whether a package is present. If this is the case, the package is picked up by a cylinder 1A1 (lifting cylinder) and then transferred to another conveyor by means of cylinder 2A1 (transfer cylinder).

Cylinder 1A1 is to retract first, followed by cylinder 2A1. The cylinders are retracted and advanced by means of solenoid valves (coils 1Y1 and 2Y1). The cylinder positions are monitored by means of proximity switches 1B1 to 2B2.

On the feed side, packages have been previously arranged in such a way that they reach the lifting device individually.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task in function chart to IEC 60848.
3. Declaration of the PLC program variables
4. Formulation of the program into a sequential function chart
   - Programming transition conditions directly in one of the languages FBD, LD or ST
   - Specifying actions as boolean actions
5. Testing and commisioning of the PLC program system.

Implementation

15.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
2	Double-acting cylinder
2	5/2-way single solenoid valve
1	Proximity switch, optical
4	Proximity switch, inductive-magnetisch
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air supply at service unit!

- Establish the electrical and pneumatic connections.

15.2 Describing the control task in function chart to IEC 60848

- Create the program in function chart to IEC 60848.

15.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

15.4 Formulation of the PLC program into the sequential function chart

- Design the sequence structure consisting of steps and transitions.
- Program the transition conditions directly in one of the languages FBD, LD or ST.
- Formulate the actions associated with the steps directly as Boolean actions.
- Create the step structure by mapping the steps onto storage elements if the sequential function chart is not supported by your PLC.

15.5 Testing and commisioning of the PLC program system

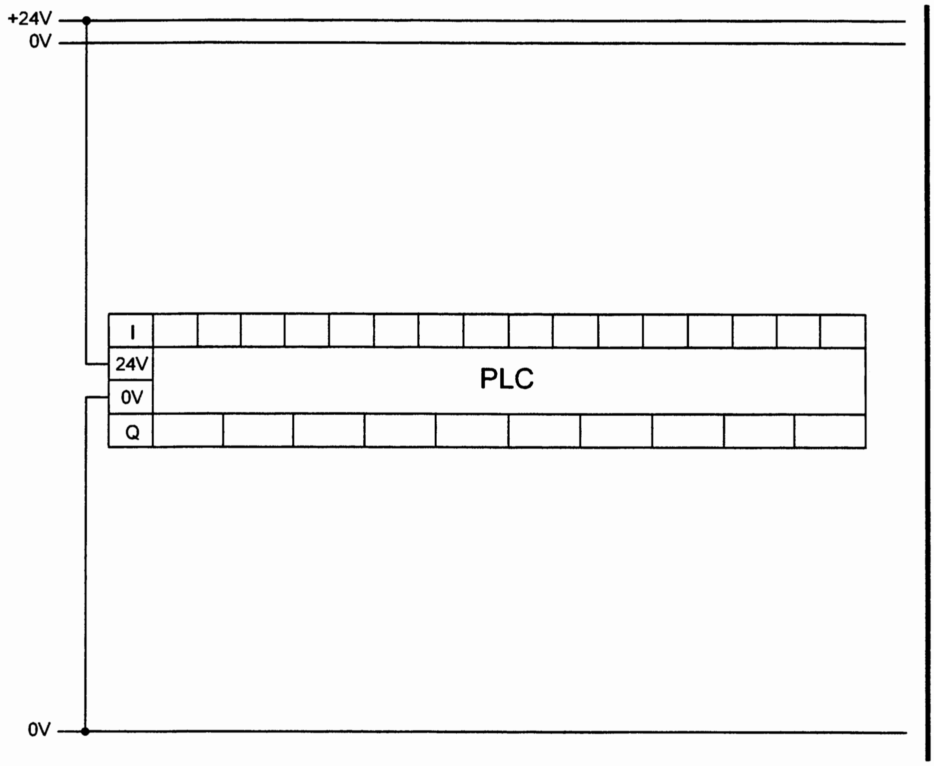
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

15.1 Drawing up the electrical circuit diagram and constructing the circuit

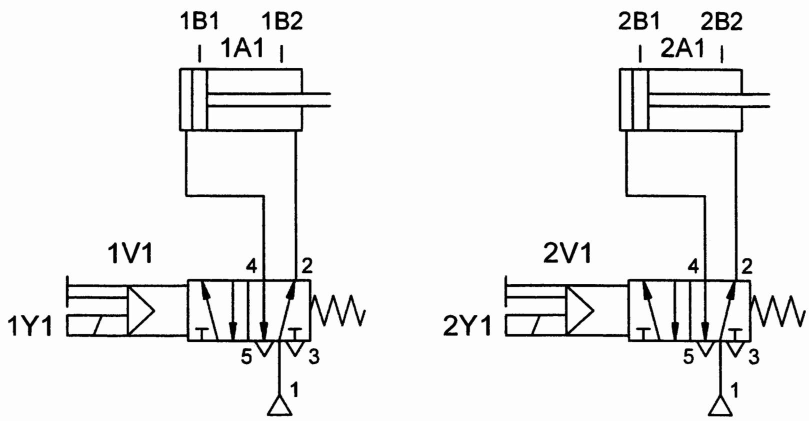
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



15.2 Describing the control task in function chart in accordance with IEC 60848

Create the program in function chart in accordance with IEC 60848.

WORKSHEET

15.3 Declaration of the PLC program variables

Declare the variables reqired in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

15.4 Formulation of the PLC program into a sequential function chart

PLC program in sequential function chart

WORKSHEET

Question 1:

What is the function of a step without associated actions?

Question 2:

What is the response of the sequential function chart program if the action of step S3: "Cylinder 2A1 to advance" is programmed as a non-stored action?

Subject

Programmable logic controllers

Title

Lifting and sorting device for packages
Alternative branching

Training aim

- To be able to program a sequence control system with alternative branching

Technical knowledge

Sequence control system with alternative branching

There are sequence control systems, where several different sequences may occur. A sequence is selected depending on the signals originating from the process applied.

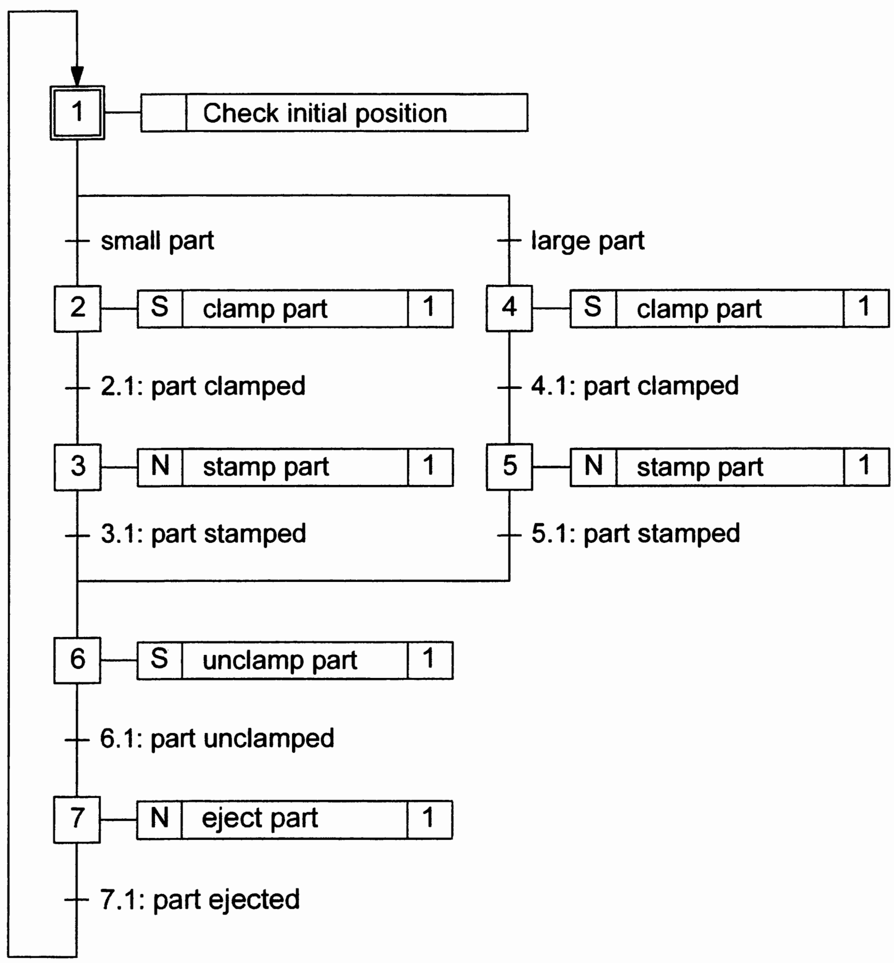


Fig. A16.1:
Example of a sequence control system with alternative branching

A stamping tool, which stamps small or large parts by means of two different cylinders, represents an example of such a sequence control system.

Fig. A16.1 illustrates the function chart to IEC 60848 for the above mentioned example.

Alternative branching is represented by as many transitions below the horizontal line as there are different sequences possible.

In order to select one option only, the transition conditions must be mutually exclusive.

Two sequences are available for selection in the example given. If a small part is detected, only steps 1, 2, 3, 6 and 7 are processed. If a large part is present, the program branches to step 4, 5, 6 and 7 after step 1.

Programming of a sequence control system with alternative branching

Sequence control systems represented in function chart to IEC 60848 are very easily programmed in sequential function chart. In the program listed below, the transition conditions have been formulated in the language ST.

The initial step S1 is active after the start of the program. S1 is a void step in this instance, i.e. no actions have been assigned to this step.

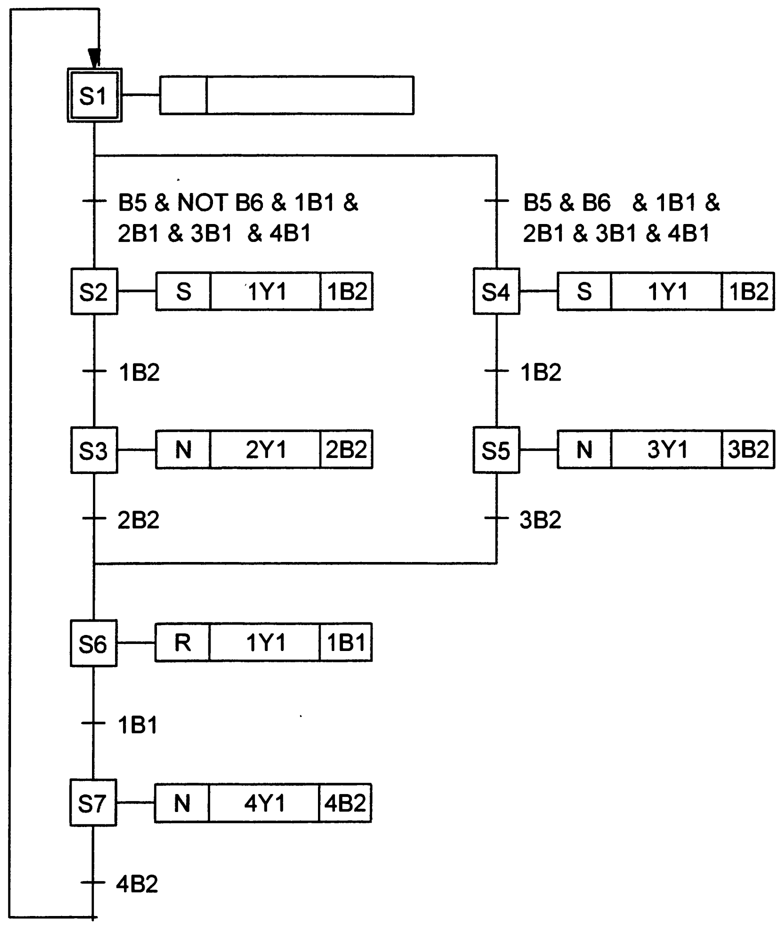
If all the cylinders are retracted and a small part is present, step S2 is set and step S1 reset. The part is then clamped, stamped, declamped and finally ejected.

If a large part has been detected (B5=1 and B6=1), steps S1, S4, S5, S6 and S7 are executed consecutively.

Following this, processing of the steps starts with step S1 again.

Fig. A16.2:
Example of a sequential function chart with alternetive branching

```
VAR
   B5   AT  %IX1.0   :  BOOL;    (* small or large part        *)
   B6   AT  %IX1.1   :  BOOL;    (* for large part only        *)
   1B1  AT  %IX2.0   :  BOOL;    (* Cylinder 1A1 retracted     *)
   1B2  AT  %IX2.1   :  BOOL;    (* Cylinder 1A1 extended      *)
   2B1  AT  %IX2.2   :  BOOL;    (* Cylinder 2A1 retracted     *)
   2B2  AT  %IX2.3   :  BOOL;    (* Cylinder 2A1 extended      *)
   3B1  AT  %IX2.4   :  BOOL;    (* Cylinder 3A1 retracted     *)
   3B2  AT  %IX2.5   :  BOOL;    (* Cylinder 3A1 extended      *)
   4B1  AT  %IX2.6   :  BOOL;    (* Cylinder 4A1 retracted     *)
   4B2  AT  %IX2.7   :  BOOL;    (* Cylinder 4A1 extended      *)
   1Y1  AT  %QX1.0   :  BOOL;    (* Cylinder 1A1: clamping     *)
   2Y1  AT  %QX1.1   :  BOOL;    (* Cylinder 2A1: stamping small *)
   3Y1  AT  %QX1.2   :  BOOL;    (* Cylinder 3A1: stamping large *)
   4Y1  AT  %QX1.3   :  BOOL;    (* Cylinder 4A1: ejecting     *)
END_VAR
```



In the above example, the actions for steps S3, S5 and S7 are programmed as non-stored. This is indicated by the qualifier N. The boolean variables 2Y1, 3Y1 and 4Y1 therefore only carry a 1-signal while the associated steps are active.

If the sequential function chart is not available for programming, the step sequences may also be generated in this instance by means of using storage elements.

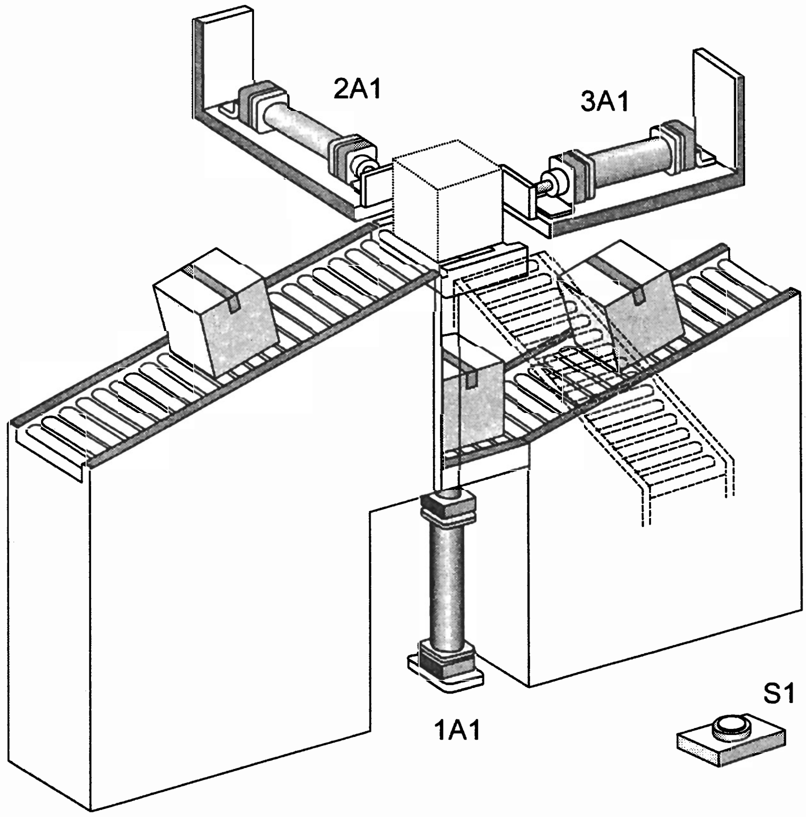
Problem description

Packages are conveyed past a linear measuring device on a roller conveyor in order to establish their size. There are two different package sizes: Short and long packages. The linear measuring device supplies an 0-signal for a short package and a 1-signal for a long package.

Following this, the packages reach a lifting platform. The sequence is started by means of START button S1. The packages are lifted by a lifting cylinder 1A1. The packages are then sorted: Short packages are transferred to a second conveyor via transfer cylinder 2A1, and long packages onto a third conveyor via cylinder 3A1. Lifting cylinder 1A1 is to retract again only after cylinders 2A1 or 3A1 have reached their end position.

The cylinder positions are monitored by means of proximity switches 1B1 to 3B2. Cylinder 1A1 is advanced and retracted by means of a double solenoid valve via coils 1Y1 and 1Y2. Cylinders 2A1 and 3A1 are advanced and retracted by means of solenoid valves (coils 2Y1 and 3Y1).

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
6. Describing the control task in function chart to IEC 60848.
7. Declaration of the PLC program variables
8. Formulation of the program into a sequential function chart
   - Programming transition conditions directly in one of the languages FBD, LD or ST
   - Specifying actions as boolean actions
9. Testing and commisioning of the PLC program system.

Implementation

16.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
2	Double-acting cylinder
2	5/2-way single solenoid valve
1	5/2-way double solenoid valve
1	Signal input, electrical
1	Proximity switch, capacitive
1	Proximity switch, optical
4	Proximity switch, inductive
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air supply at service unit!

- Establish the electrical and pneumatic connections.

16.2 Describing the control task in function chart to IEC 60848

- Create the program in function chart to IEC 60848

16.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

16.4 Formulation of the PLC program in sequential function chart

- Design the sequence structure consisting of steps and transitions.
- Program the transition conditions directly into one of the languages FBD, LD or ST.
- Formulate the actions associated with the steps directly as Boolean actions.
- Create the step structure by mapping the steps onto a storage element, if the sequential function chart is not supported by your PLC.

16.5 Testing and commisioning of the PLC program system

- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

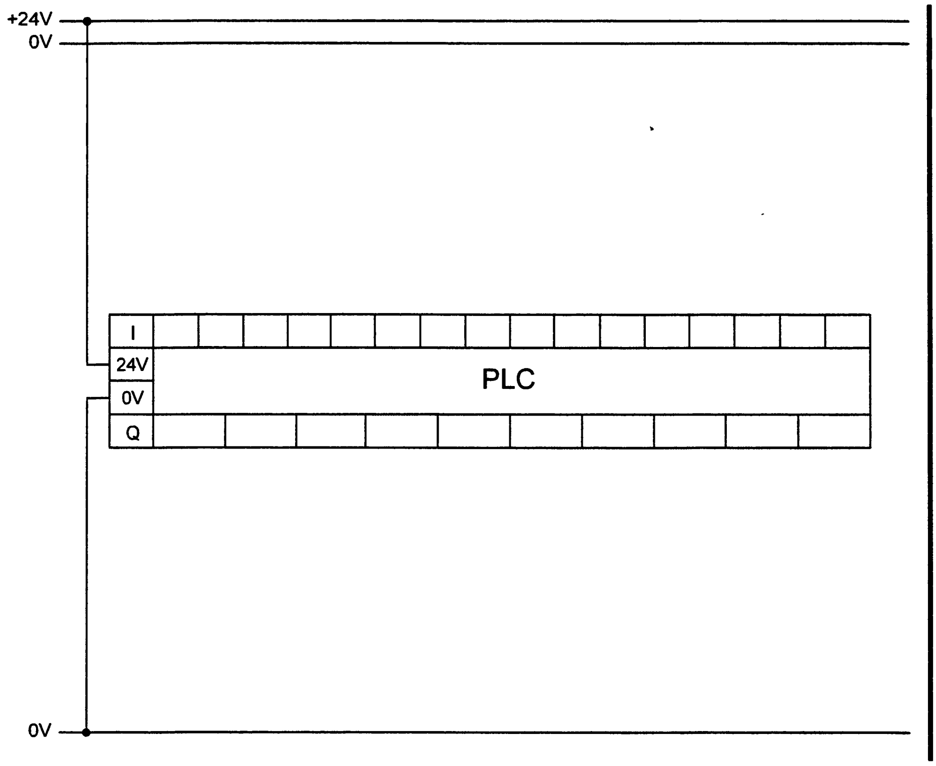
- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

16.1 Drawing up the electrical circuit diagram and constructing the circuit

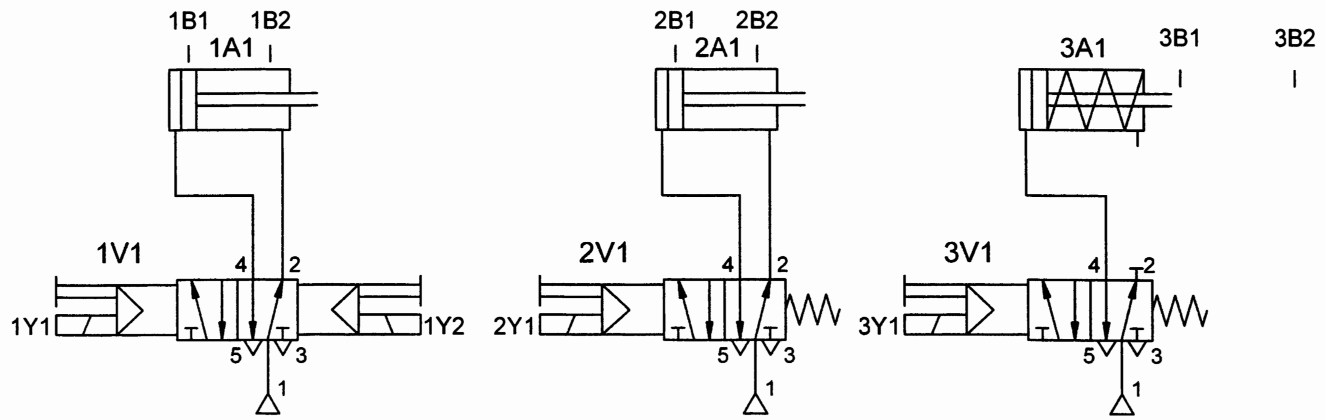
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.

Circuit diagram, electrical



Configure the control system.

Circuit diagram, electro-pneumatic



16.2 Describe the control task in function chart to IEC 60848

Create the program in function chart to IEC 60848.

Question 1:

What is the sorting criteria according to which the packages are evaluated?

WORKSHEET

Question 2:

How do you ensure that just one sequence step is selected during program execution?

16.3 Declaration of the PLC program variables

Declare the variables reqired in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

16.4 Formulate the PLC program in sequential function chart

PLC program in sequential function chart

Question:

Specify the transition condition, which is always true. Why are such transition conditions formulated?

Subject

Programmable logic controllers

Title

Stamping device with counter
Counting cycles

Training aim

- To be able to realise counting cycles by means of using the standard function modules CTU or CTD

Technical knowledge

Counting cycles form part of the basic operations of a PLC. EN 61131-3 (IEC 61131-3) defines the three standard function blocks CTU (incremental counter), CTD (decremental counter) and CTUD (incremental/decremental counter) for the realisation of these operations.

Function block CTU, incremental counter

Function block CTU (Fig. A17.1) realises an incremental counter. Its interface is defined by means of three input and two output parameters.

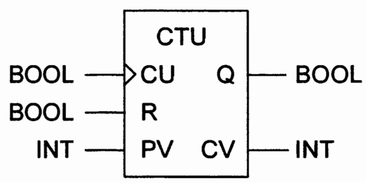


Fig. A17.1:
Function block CTU

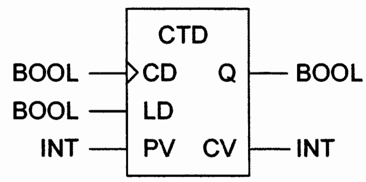
The characteristic behaviour of an incremental counter is as follows:

- A 1-signal at reset input R sets the current counter status CV at 0.
- Thereafter, the value CV of the counter is increased by 1 with each rising edge at input CU.
- A 1-signal applies at output Q as soon as the current value CV is equal or greater than the preselect value PV. Output Q carries a 0-signal as long as current counter status CV is less than the preselect value PV.

Function block CTD, Decremental counter

Function block CTD (Fig. A17.2) being a decremental counter is counterpart to function block CTU.

Fig. A17.2:
Function block CTD



The behaviour of a decremental counter is as follows:

- A 1-signal at the LD input sets the current counter status CV equal to the preselect value PV.
- Each rising edge at the CD input decreases the current counter status CV by 1.
- Output Q carries a 0-signal as long as the current counter status CV is greater than 0. Only if the current value is less or equal to 0, does a 1-signal apply at output Q.

Use of function block CTU in the individual programming languages

The use of the function block is demonstrated on a small packaging task.

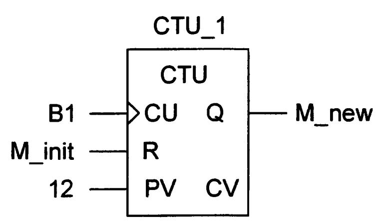
Example

12 parts each are to be packed into one box. When a box has been filled, another is made available. Each packaging cycle is triggered via a memory M_init. The individual parts are detected by means of a sensor B1. The status of the counter is mapped onto a memory M_new.

Table A17.1: Application of an incremental counter

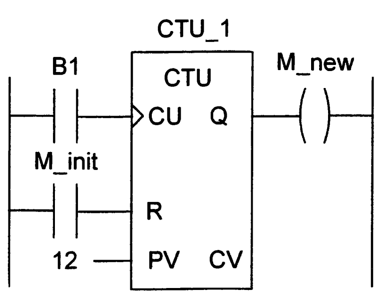
```
VAR
  B1      AT %IX1.0 : BOOL;     (* detects part to be packaged  *)
  M_init  AT %MX1.0 : BOOL;     (* detects whether counter      *)
                                (* needs to be initialised      *)
  M_new   AT %MX1.1 : BOOL;     (* detects whether new box      *)
                                (* is required                  *)
  CTU_1             : CTU;      (* Incremental counter named    *)
                                (* CTU_1 for counting cycle     *)
END_VAR
```

FBD



Function block CTU_1, connected with the current parameters.

LD



Interconnection of function block CTU_1 in the rung.

Table A17.1
Application of an incremental counte (continuation)

IL

`CAL CTU_1 (CU := B1, R := M_init, PV := 12)`	Invocation of function block CTU_1
`LD  CTU_1.Q`	Reading of output Q of CTU_1
`ST  M_new`	Storage of current results to M_new

ST

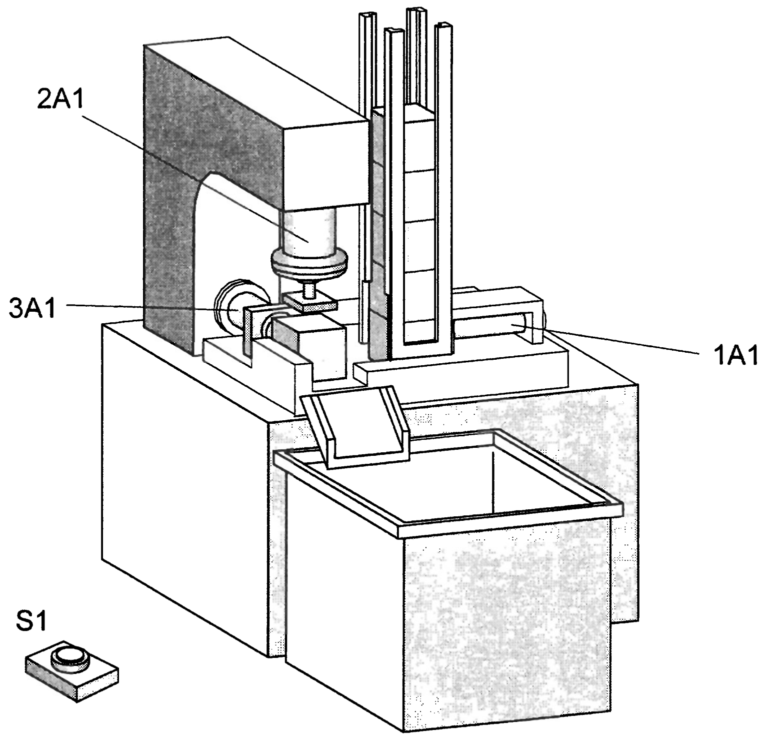
`CTU_1 (CU := B1, R := M_init, PV := 12);`	Invocation of function block CTU_1
`M_new := CTU_1.Q;`	Assignment of output Q of CTU_1 to M_new

Problem description

10 parts at a time are stamped on a machine. The program cycle is started by means of a push button S1. The proximity switch B4 signals "Part in magazine". A part is fed towards the machine by means of a cylinder 1A1 and clamped. It is then stamped via cylinder 2A1, and subsequently ejected by means of an ejecting cylinder 3A1.

The clamping cylinder 1A1 operates via a double solenoid valve with two coils 1Y1 (clamping) and 1Y2 (unclamping). Cylinders 2A1 and 3A1 are powered by spring-return solenoid valves with the coils 2Y1 and 3Y1. The cylinder positions are monitored by means of the proximity switches 1B1 to 3B2.

Positional sketch



Exercise definition

1. Drawing up the electrical circuit diagram and constructing the circuit
2. Describing the control task in function chart to IEC 60848.
3. Declaration of the program variables
4. Formulation of the program in sequential function chart
   - Programming the transition conditions in one of languages FBD, LD or ST
   - Specifying the actions
5. Testing and commisioning of the PLC program system.

Implementation

17.1 Drawing up the electrical circuit diagram and constructing the circuit

- Complete the electrical circuit diagram on the worksheet.
- Assemble the required equipment on the slotted assembly board:

Components list

Quantity	*Description*
1	Programmable logic controller
1	Interconnecting cable for connection unit
1	Connection unit
1	Service unit
1	Manifold
1	Quick push-pull distributor
1	Single-acting cylinder
2	Double-acting cylinder
2	5/2-way single solenoid valve
1	5/2-way double solenoid valve
1	Signal input, electrical
1	Proximity switch, capacitive
1	Proximity switch, optical
1	Proximity switch, inductive
4	Proximity switch, inductive-magnetisch
	Plastic tubing

- **Prior to wiring and tubing:**
  Switch off power supply!
  Switch off air suplp at service unit!

- Establish the electrical and pneumatic connections.

17.2 Describing the control task in function chart to IEC 60848

- Create the program in function chart to IEC 60848.

17.3 Declaration of the PLC program variables

- All variables are to be created as program-local variables.
- Specify only those parts of the declaration required for your PLC application. These are: Designation, data type, address – only if directly addressed variables are used – and variables comment.

The component parts of the declaration of variables in this exercise section is represented in tabular form. If actual PLC systems are used, the input and representation of the variables declaration is dependent on the PLC programming system used.

17.4 Formulate the PLC program in sequential function chart

- Design the sequence structure consisting of steps and transitions.
- Program the transition conditions directly in one of the languages FBD, LD or ST.
- Formulate the actions associated with the steps. For actions consisting of more than once boolean variable, it is mandatory to input an action name.
- Create the step structure by mapping the steps onto storage elements if the sequential function chart is not supported by your PLC.

17.5 Testing and commisioning of the PLC program system

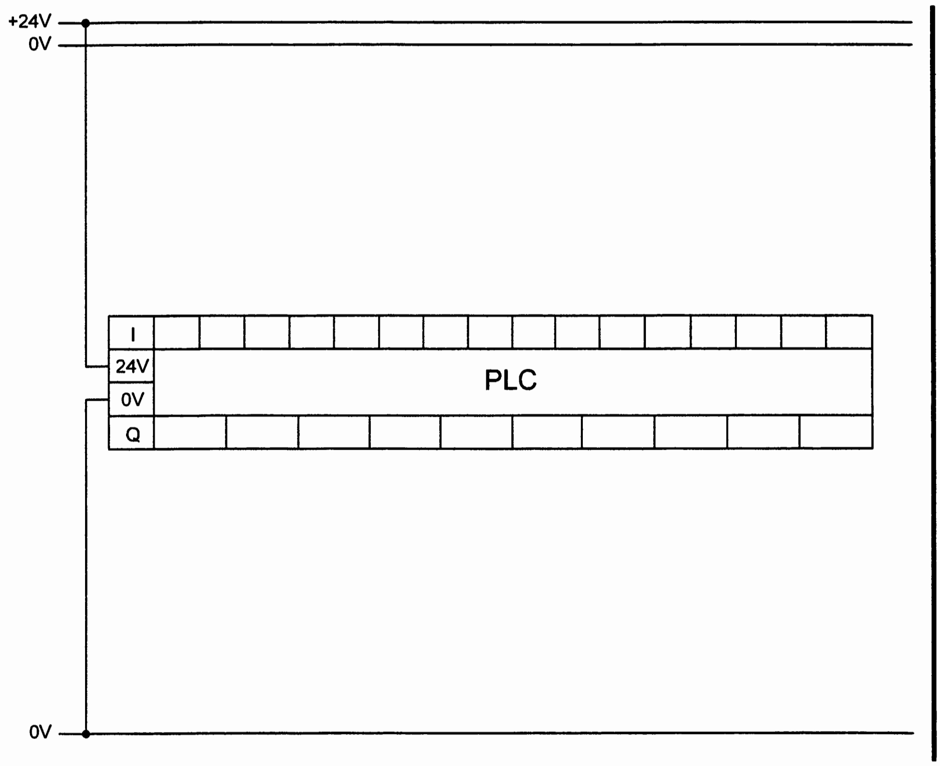
- **Prior to commissioning of the installation:**
  Check the assembled circuit with the help of the circuit diagrams!
- **Commissioning of the installation:**
  Switch on power supply using a standard voltage of 24 V DC!
  Increase air supply on service unit to operating pressure (see data sheet for pneumatic components)!
- **Operation of the installation:**
  Keep clear of the operational parts of the installation!

- Load the program to the PLC.
- Carry out a function check.
- Correct any errors occurring in the PLC program.
- Document your solution.

WORKSHEET

17.1 Drawing up the electrical circuit diagram and constructing the circuit

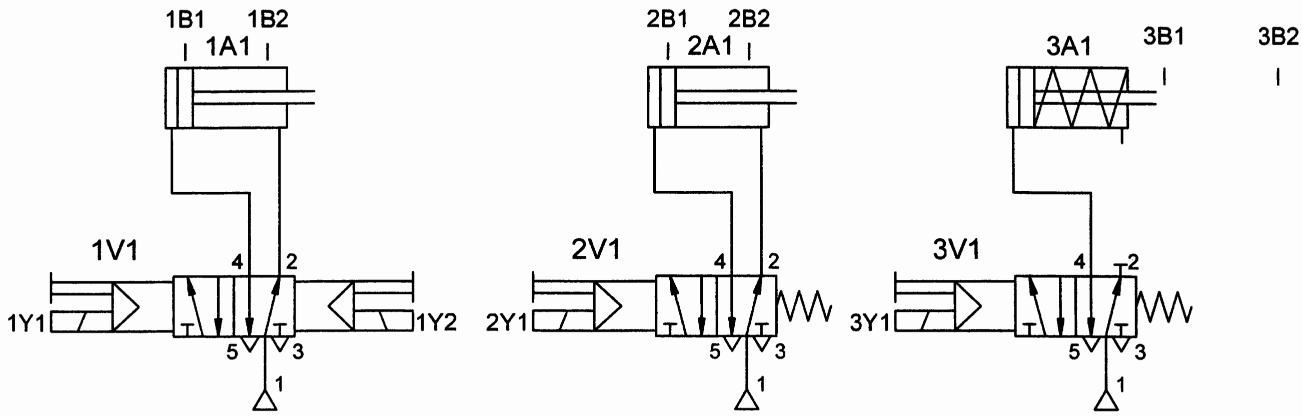
Complete the electrical circuit diagram and enter the input and output addresses available for your PLC.



Circuit diagram, electrical

Configure the control system.

Circuit diagram, electro-pneumatic



17.2 Describing the control task in function chart to IEC 60848

Create the program in function chart to IEC 60848.

WORKSHEET

17.3 Declaration of the PLC program variables

Declare the variables required in your PLC program:

Declaration of variables

Designation	*Data type*	*Address*	*Comment*

17.4 Formulate the PLC program in sequential function chart

PLC program in sequential function chart

Question:

When does the status of a counter change?