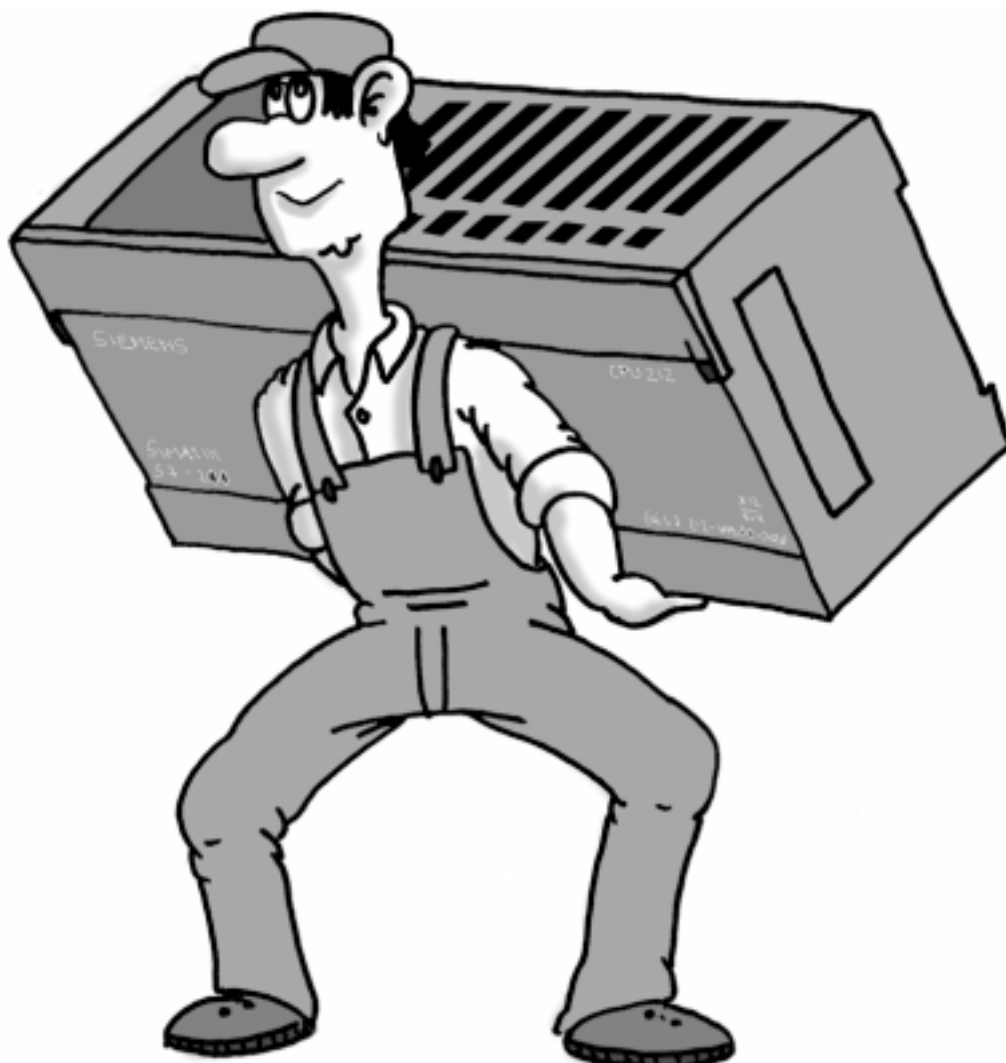


# Contents of the S7-200 Starter Kit



Item	Quantity	available
S7-200 CPU 221 Relay	1	
Simulator for CPU 221	1	
Software STEP 7-Micro/WIN 32 (V 3)	1	
Training Model on Mounting Rail	1	
PC/PPI Cable	1	
S7-200 Documentation	1	
One Hour Primer incl. Exercise Diskette	1	
Screw driver	1	

**Safety Guidelines**

The One Hour Primer was created as a quick introduction to the world of S7-200 and has deliberately been kept short. It is not a substitute for the S7-200 manual.

Therefore, please observe the instructions given in the S7-200 manual, especially the safety guidelines.

---

**Trademarks**

SIMATIC® and SIMATIC NET® are registered trademarks of Siemens AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Copyright © Siemens AG 1999 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Siemens AG  
Automation and Drives  
Industrial Automation Systems Division  
PO Box 4848, D-90327 Nuremberg  
Federal Republic of Germany

**Disclaimer of Liability**

We have checked the content of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

© Siemens AG 1999

Subject to change without prior notice

# Preface

Dear S7-200 User,

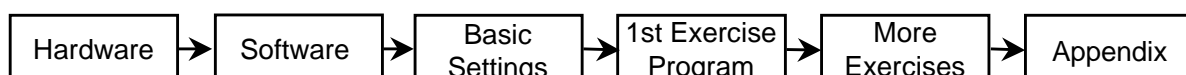
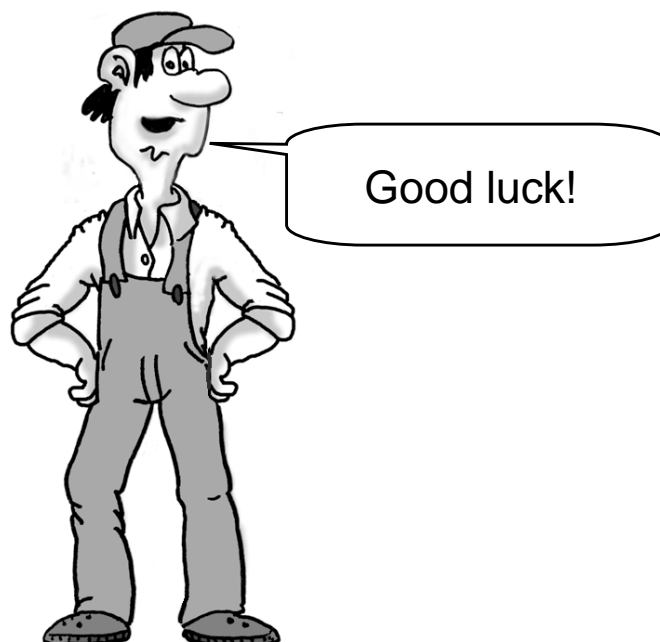
The use of programmable controllers (PLC) in automation is constantly on the increase as the pressure to reduce costs in the production process rises. The fast pace of technical development is another reason why more and more automation problems are being solved with programmable controllers.

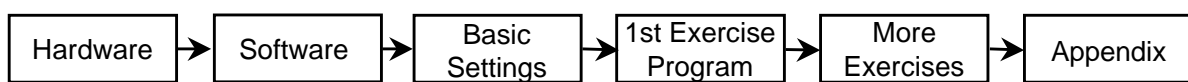
The Micro PLC S7-200, in particular, is being used for more and more applications because it combines power with an attractive price and simple operation.

To make your first steps in the world of S7-200 as simple as possible, we have created a special Starter Kit.

This One Hour Primer is intended to help you learn enough basic skills to be able to use the S7-200 within the shortest possible time.

And now, we wish you a simple and quick start and every success.



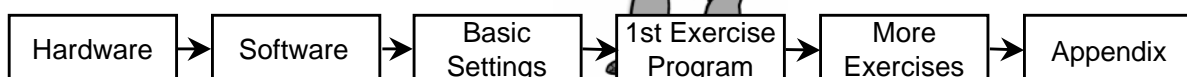


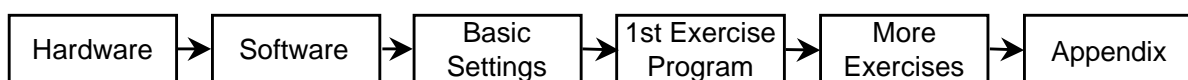
# Table of Contents

Hardware	Setting up the Hardware (Assembly)	5
	Structure of an S7-200 (CPU 221) PLC	6
	Wiring of the Training Assembly	7
	Circuit Diagram of the Training Assembly	8
	Terminal Diagram of the S7-200 (CPU 221)	9
Software	Installing the Software on Windows 95/98/NT	11
	Starting STEP 7-Micro/WIN	12
	The Help System	13
Basic Settings	Setting the Transmission Rate	15
	Configuring the Port for the PG /PC	16
	First Function Test	17
1st Exercise Program	Writing Your First Program	19
	Opening the 1st Exercise Program	20
	Downloading the 1st Exercise Program	21
	Function and Test of the 1st Exercise Program	22
	Ladder Logic Elements	23
	Transforming a Circuit Diagram	24
	Elements of the 1st Exercise Program	25
	Status View (online)	26
	Statements	27
More Exercises	1st Program Modification: AND Logic Operation	29
	Inserting a Logic Gate	30
	Entering the Operand and Testing	31
	Deleting ...	32
	2nd Program Modification: OR Logic Operation	33
	Inserting an OR Element	34
	3rd Program Modification: On-Delay Timer	35
	The On-Delay Timer Function	36
	Programming the On-Delay Timer	37
	Plain Speaking	38
	Creating a New Project	40
	Congratulations	43
Appendix	Of Bits, Bytes and Words	48
	Address Areas of the S7-200	49
	Cyclic Program Execution in the S7-200 PLC	50

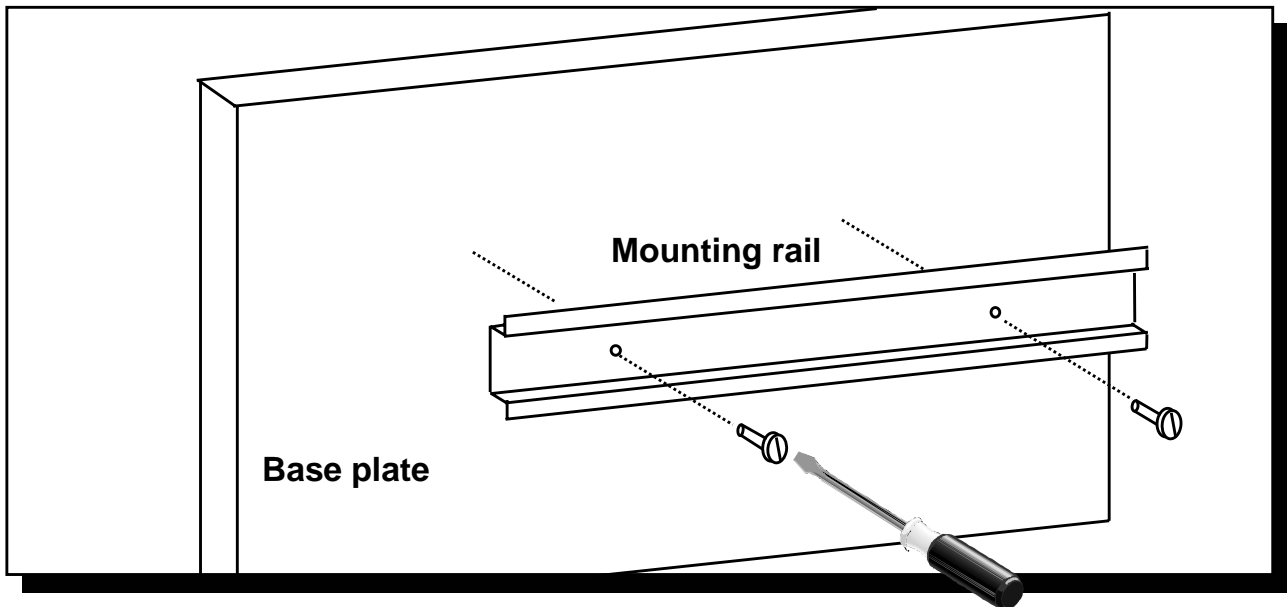


The structure of the document is shown at the foot of each page.  
The chapter which you have opened is highlighted.

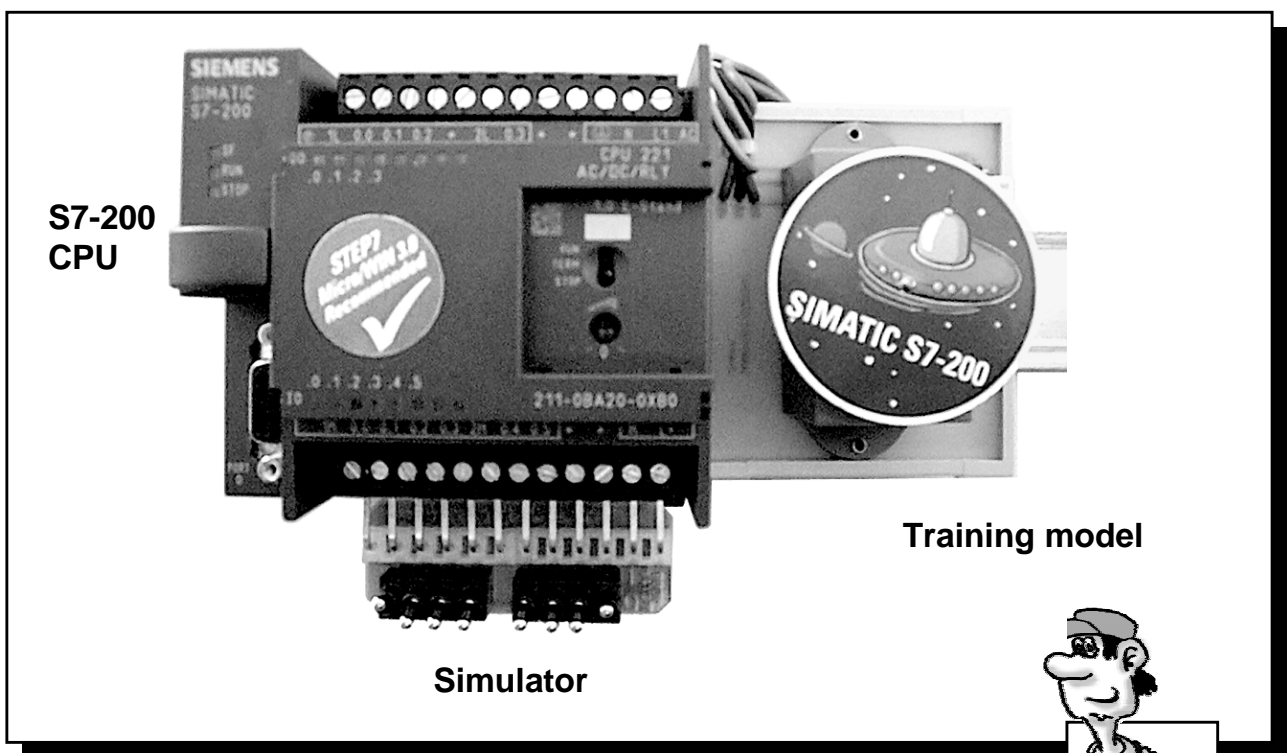




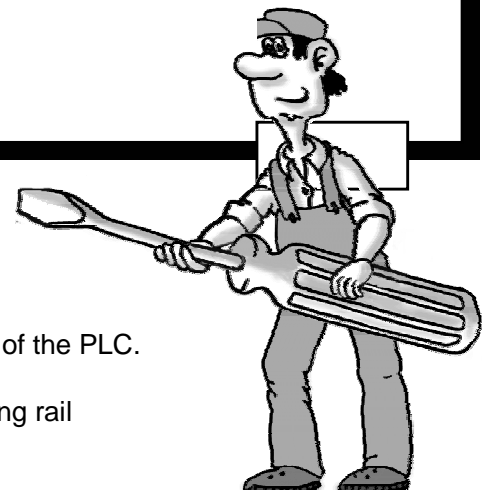
# Setting up the Hardware (Assembly)



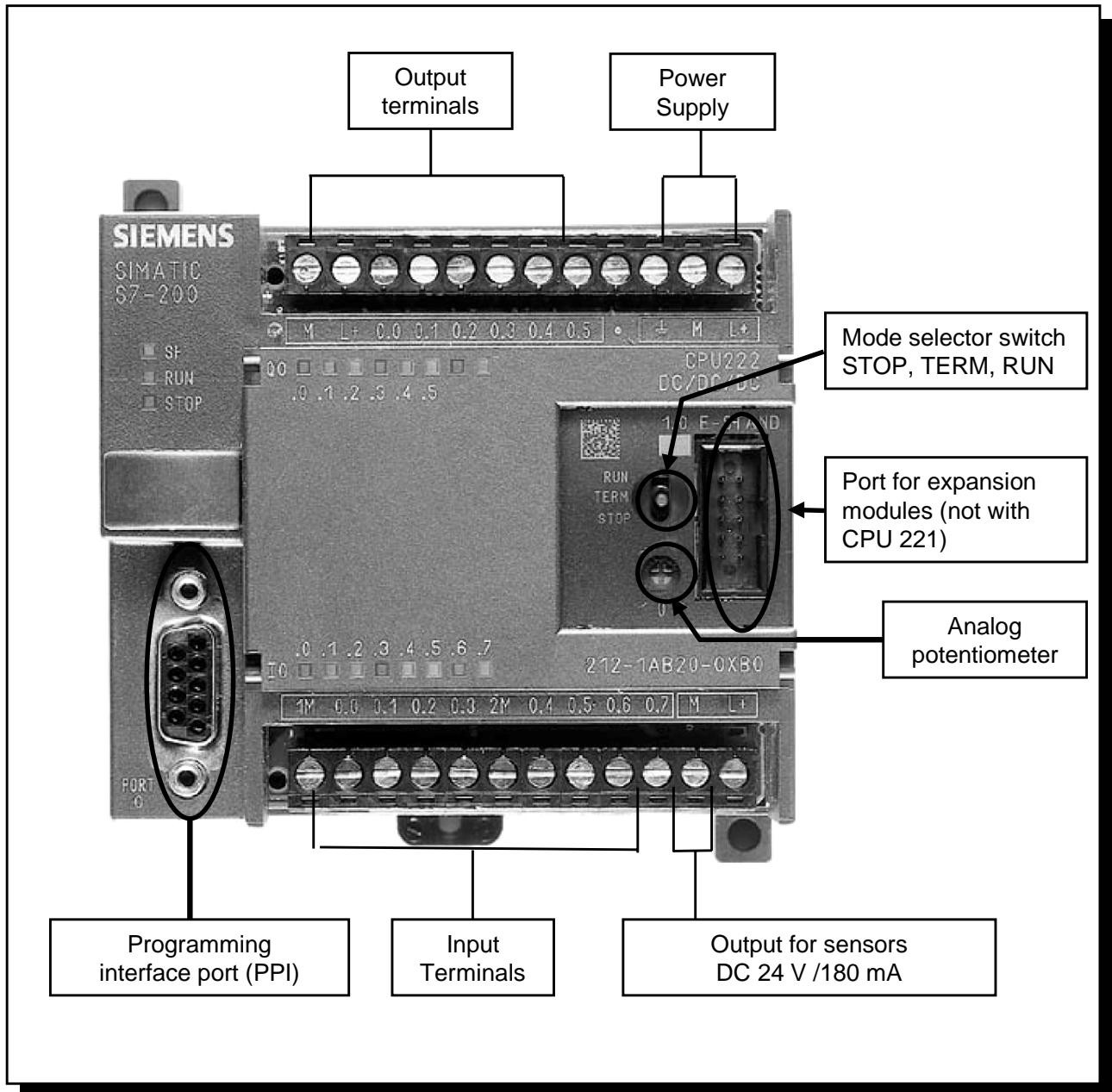
1. Mount the enclosed mounting rail on a base plate as shown in the drawing.



2. Mount the simulator on the input terminal block of the S7-200.  
The input terminal block is the terminal block on the underside of the PLC.
3. Snap the assembled PLC onto the mounting rail from above.
4. Snap the training model (part of the starter kit) onto the mounting rail next to the PLC.

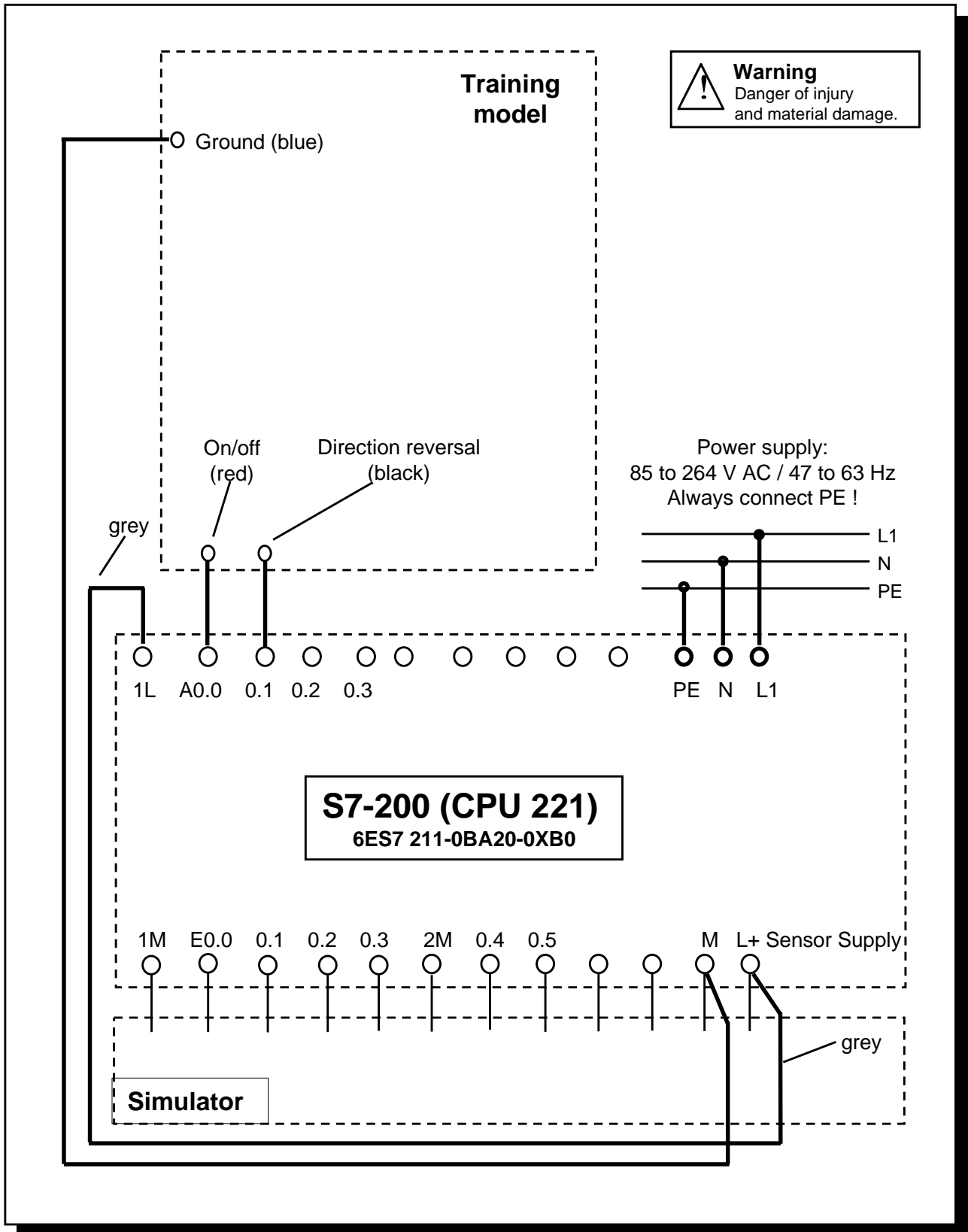


# Structure of a S7-200 (CPU 221)



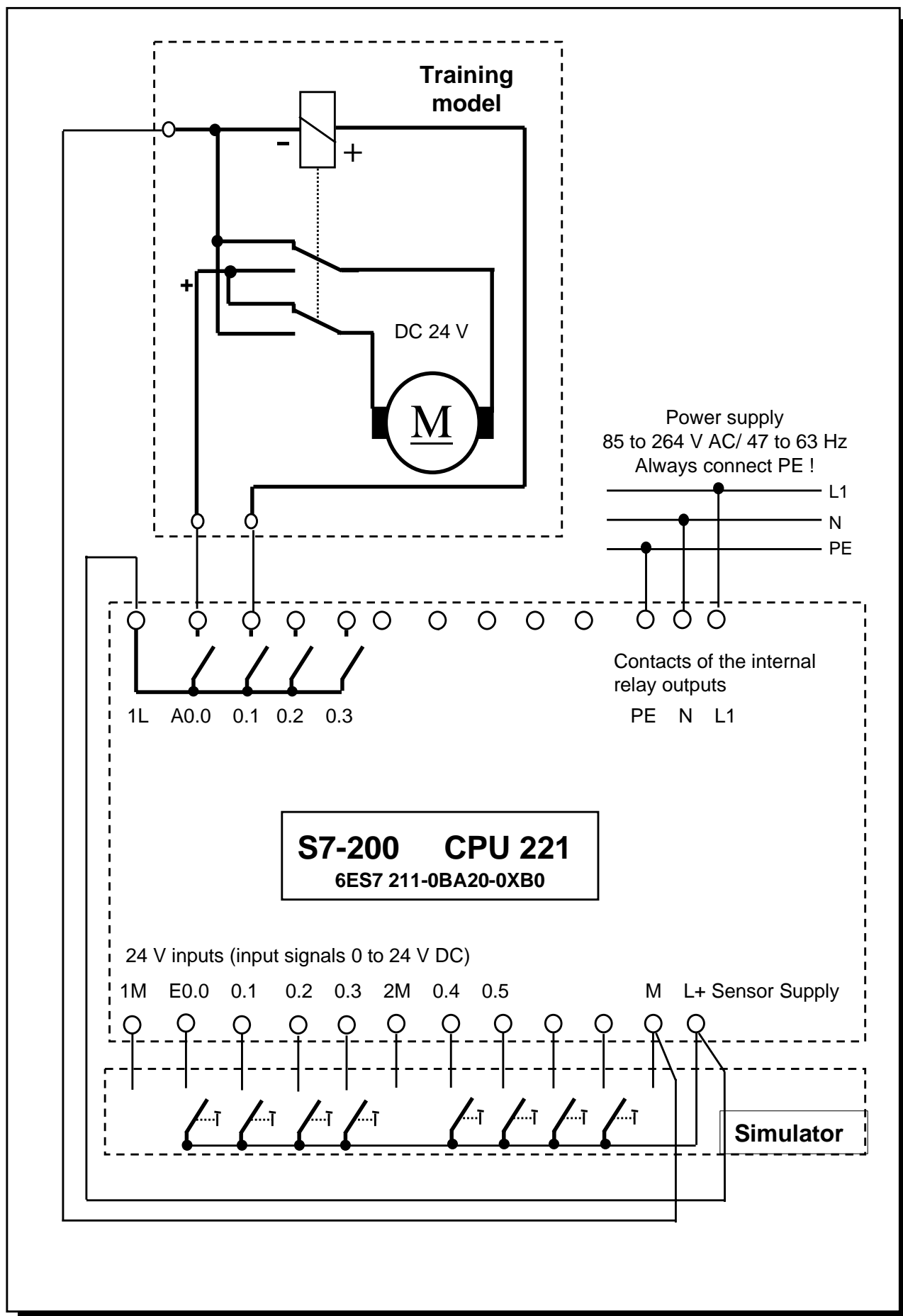


# Wiring of the Exercise Assembly

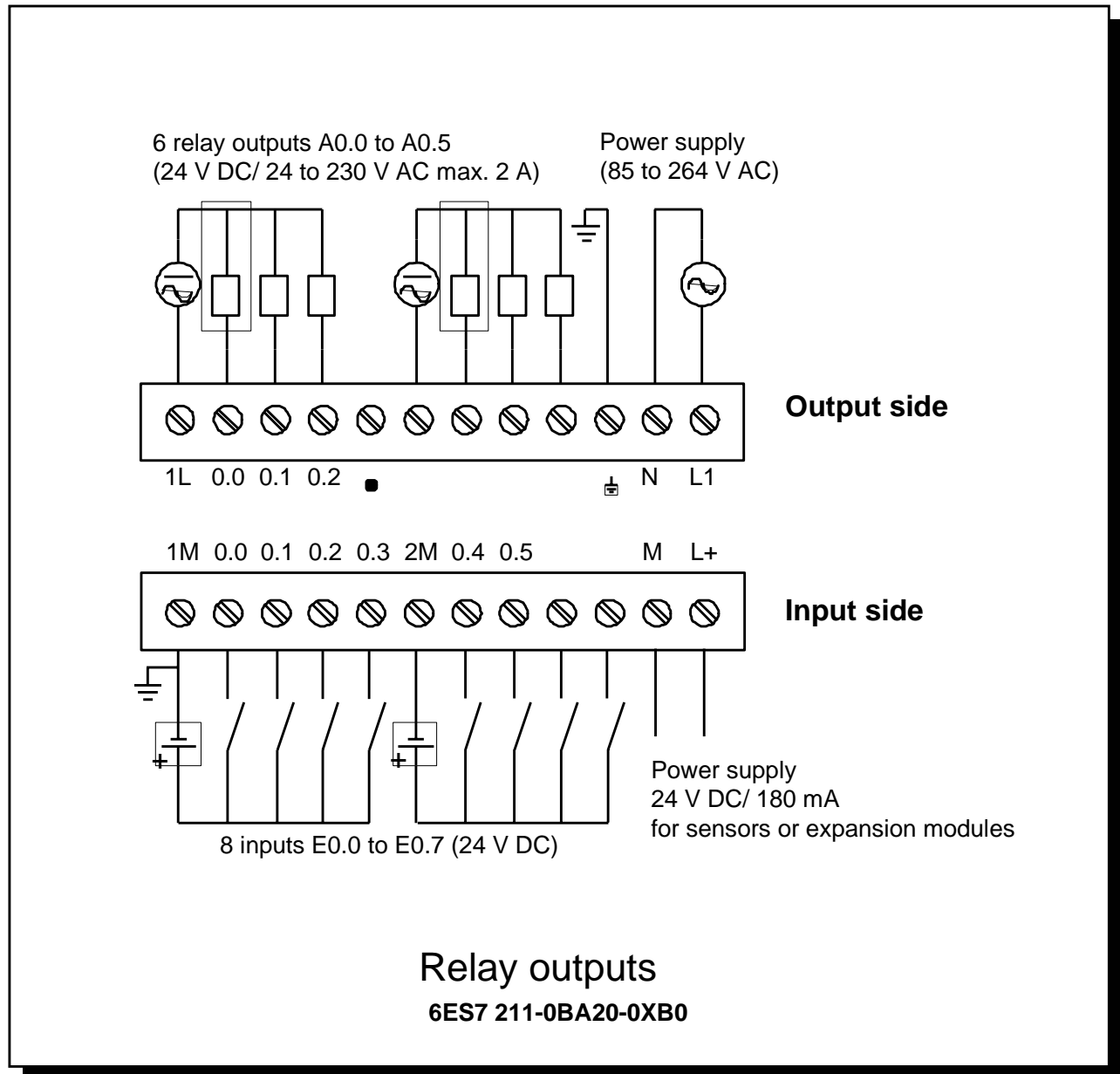


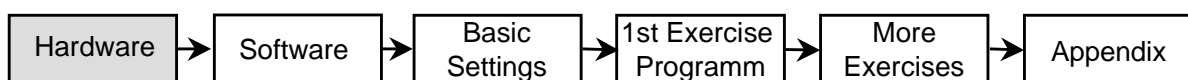
5. Connect the cables shown in boldface type as shown in the above diagram.  
 The gray wire is only fixed on the training model mechanically.  
 Any end of the gray wire can be connected to L+ and 1L.

# Circuit Diagram of the Training Assembly



# Terminal Diagram of the S7-200 (CPU 221)

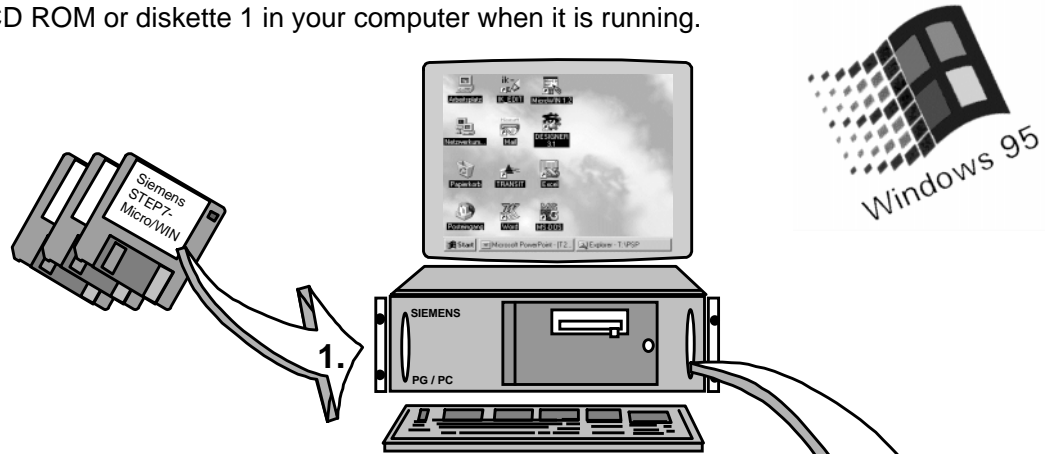




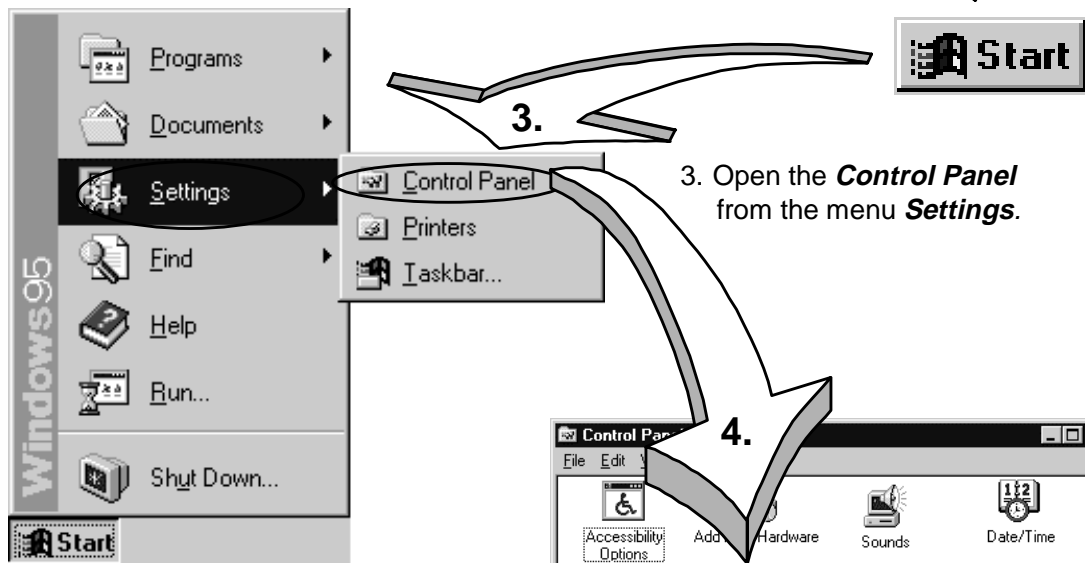
# Installing the Software on Windows 95/98/NT

In order to install the STEP 7-Micro/WIN programming software you need a PC or programming device (PG) with a Microsoft operating system. The software will run on Windows 95, Windows 98 and Windows NT 4.0.

1. Insert the CD ROM or diskette 1 in your computer when it is running.



2. Call up the start menu by selecting **Start** from the task bar.



4. Call up the software installation manager by double-clicking the mouse on the **Add/Remove Programs** icon.

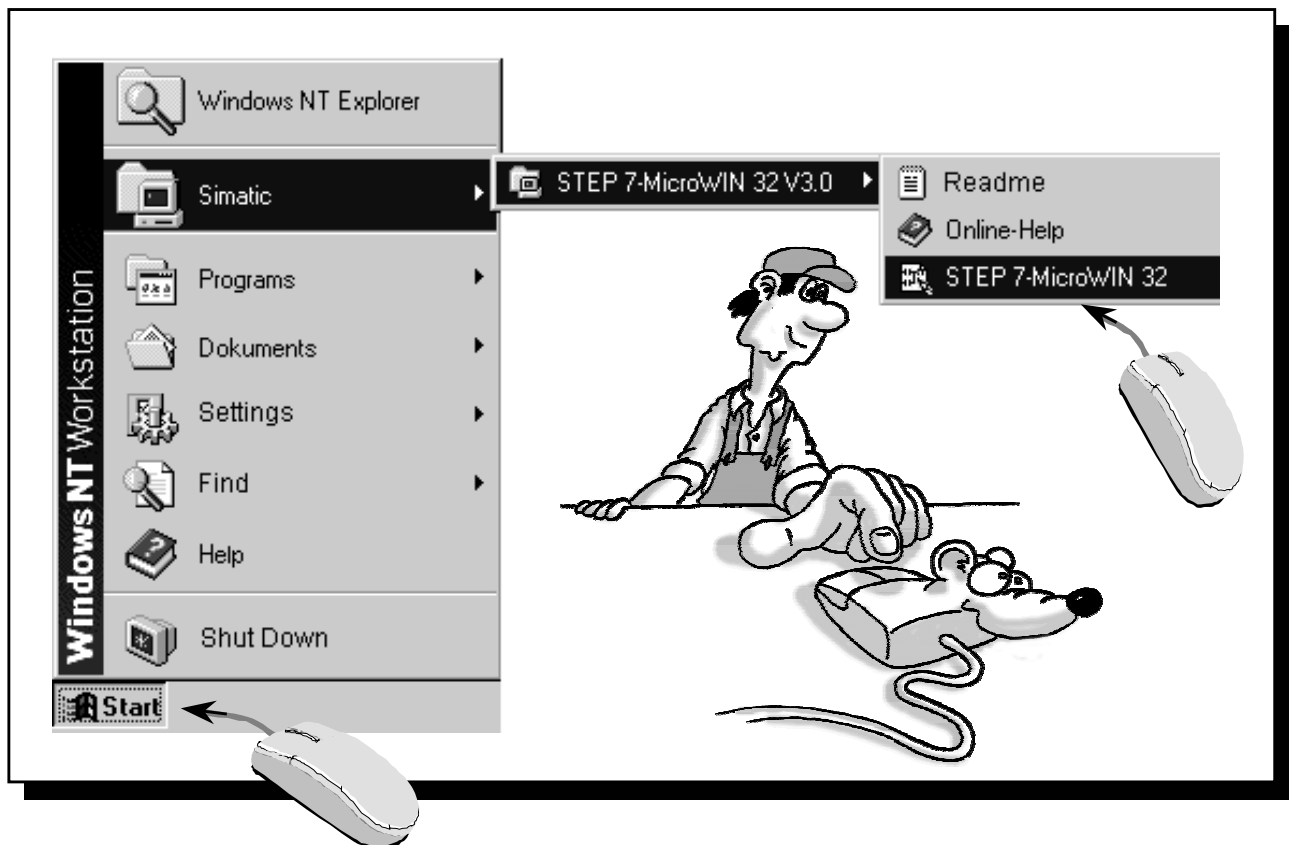
Select the **Install** button and **Continue** from the next dialog box..

The installation routine finds the setup program on STEP 7-Micro/WIN CD ROM/diskette 1.

Follow the instructions of the installation program. At the end of the installation, the entry "STEP 7-Micro/WIN 32 V3.0" appears in the menu **Start > SIMATIC**.

# Starting STEP 7-Micro/WIN

Windows 95 / 98 / NT

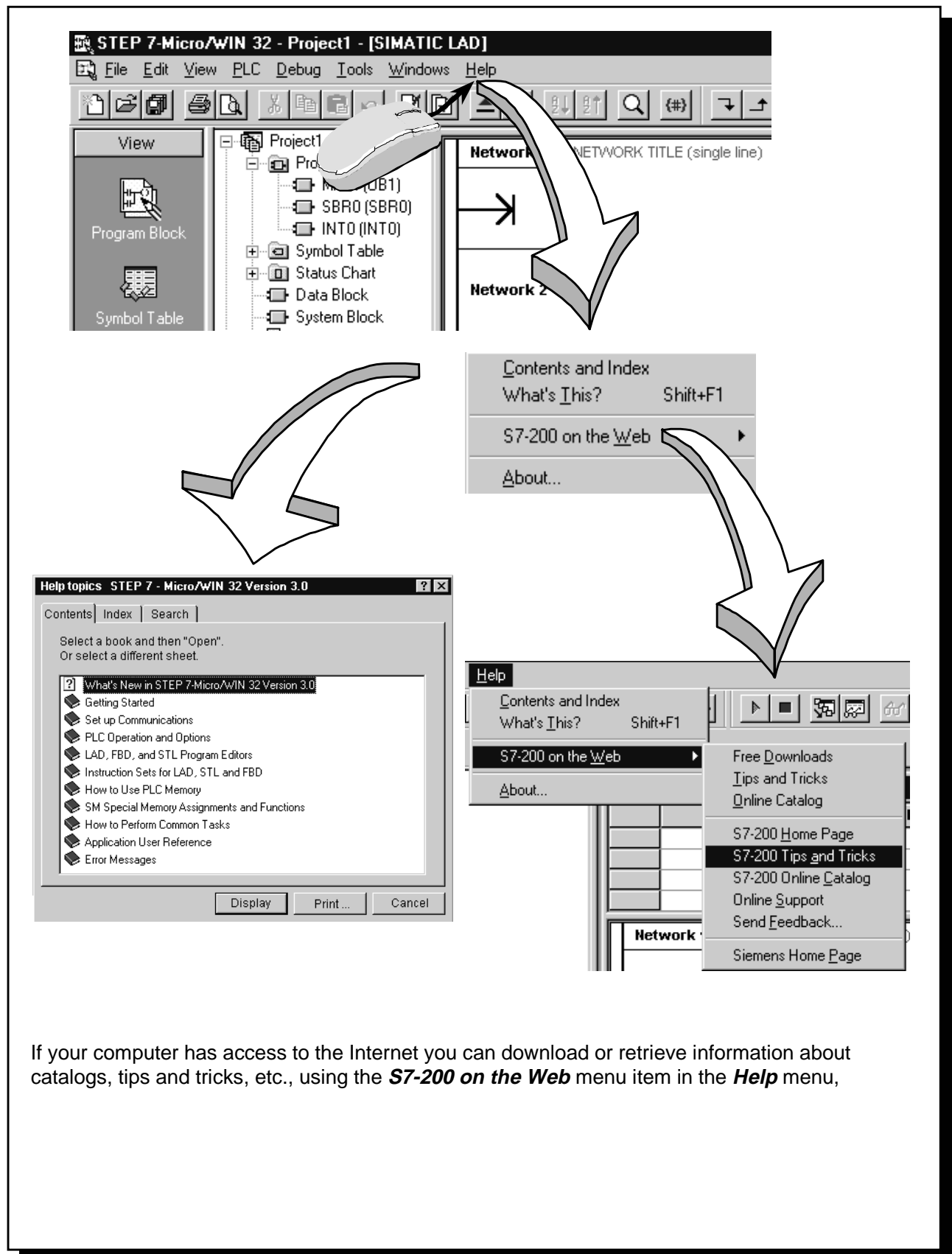


In the folder **Simatic**, which you can call up from the Start menu, you will find the folder **STEP 7-Micro/WIN 32 V3.0**. This folder contains the start icons for **STEP 7-Micro/WIN 32**.

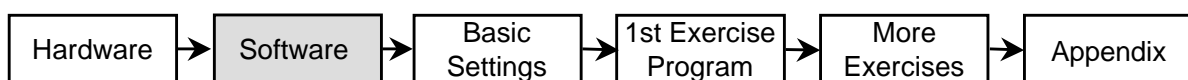
The program is started by clicking these icons.

# The Help System

STEP 7-Micro/WIN has a powerful online help system similar to other Windows applications. By selecting the menu item **Help** you can call up information on the **Contents** or the **Instruction Sets** of STEP 7-Micro/WIN.

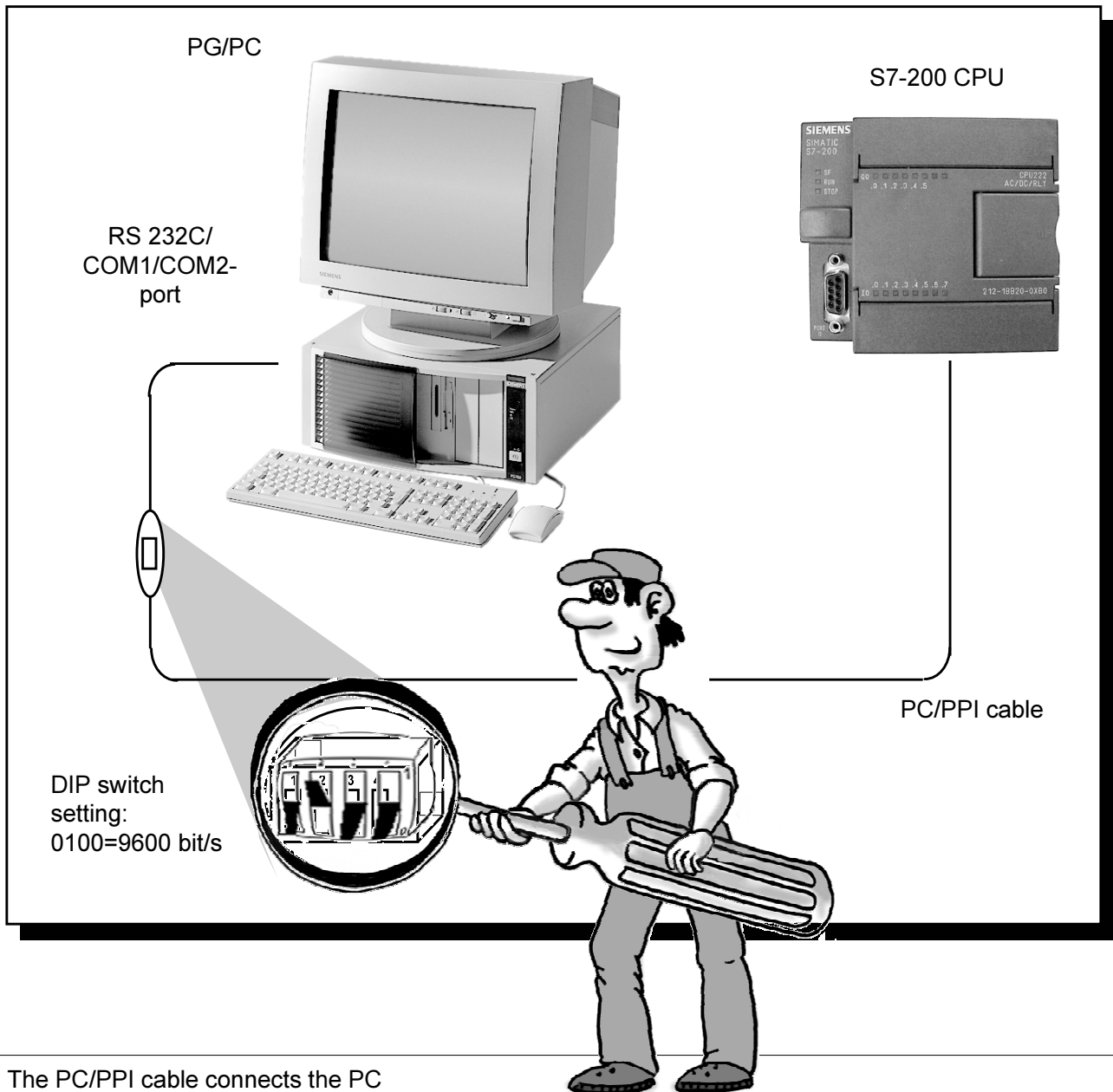


If your computer has access to the Internet you can download or retrieve information about catalogs, tips and tricks, etc., using the **S7-200 on the Web** menu item in the **Help** menu,





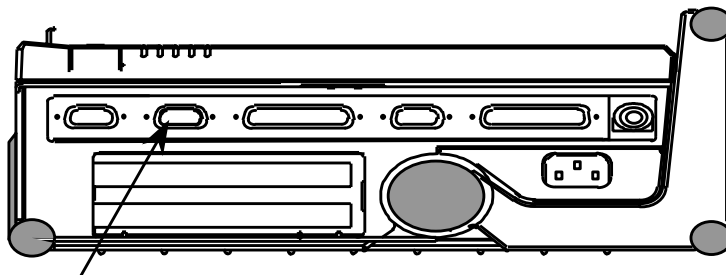
# Setting the Transmission Rate



The PC/PPI cable connects the PC to the S7-200 PLC.

On your programming device/PC you should use the serial port with the 9-pin sub D socket (or the 25-pin sub D socket with an adapter), e.g. COM 2.

The S7-200 sends and receives data at 187500 bit/s. Please refer to the diagram above to set the transmission rate on the PC/PPI. Then plug the PC/PPI cable into the PC and PLC, as shown above. Use the same phase for the power supply of the programming device/PC and the programmable logic controller (connect short end of the cable to the PC/PG), in order to avoid voltage differences. Supply the PLC with the operating voltage. (The STOP or RUN LED lights up).



COM 2 port on the PG 740 (left side)

# Configuring the Port for the PG/PC

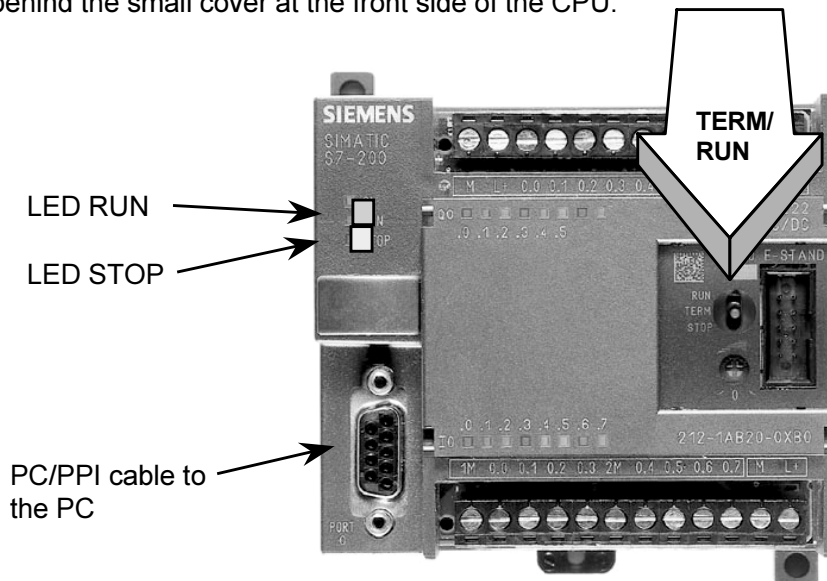
The image shows a sequence of seven numbered steps for configuring the PG/PC interface in SIMATIC Manager:

- Click to the Communications icon in the navigation bar.
- Check the communications settings.
- Refresh the communications by double-clicking the corresponding field. The connected CPU should now be recognized and registered automatically.
- If the CPU has not been recognized or if a pop-up window tells you that communication is not possible, double-click the field **PC/PPI cable**.
- Mark **PC/PPI cable** in the PG/PC Interface Setting menu and select **Properties...**
- In the PPI menu set the CPU address to 2 and the transmission rate to 9.6 kbps. In the **Local Connection** menu select the port to which you have connected the PC/PPI cable to. Confirm all settings by clicking the **OK** button.
- Refresh the communications by double-clicking again the corresponding field in the communications link menu. The CPU will now be recognized and registered automatically. The process could take a few seconds. Now close the menu for the communications links.

- Click to the Communications icon in the navigation bar.
- Check the communications settings.
- Refresh the communications by double-clicking the corresponding field. The connected CPU should now be recognized and registered automatically.
- If the CPU has not been recognized or if a pop-up window tells you that communication is not possible, double-click the field **PC/PPI cable**.
- Mark **PC/PPI cable** in the PG/PC Interface Setting menu and select **Properties...**
- In the PPI menu set the CPU address to 2 and the transmission rate to 9.6 kbps. In the **Local Connection** menu select the port to which you have connected the PC/PPI cable to. Confirm all settings by clicking the **OK** button.
- Refresh the communications by double-clicking again the corresponding field in the communications link menu. The CPU will now be recognized and registered automatically. The process could take a few seconds. Now close the menu for the communications links.

# First Function Test

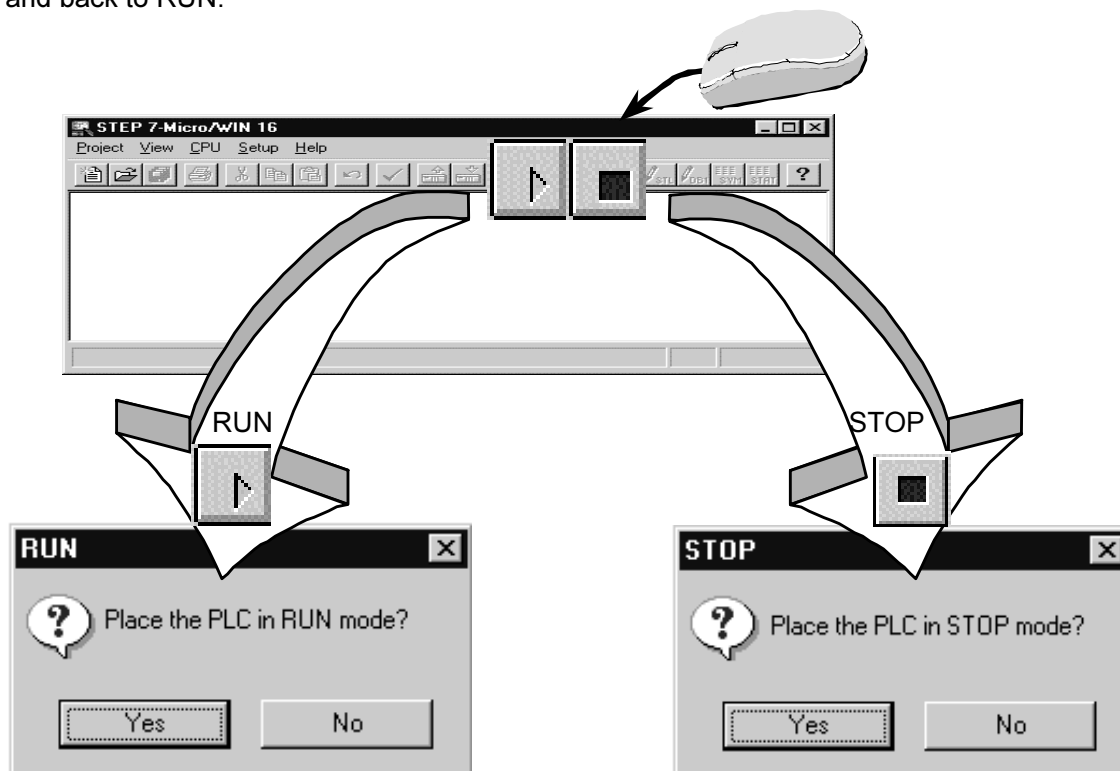
1. Set the mode selector of the PLC to the Term or RUN position. The mode selector is located behind the small cover at the front side of the CPU.



Only in the **TERM** or **RUN** position is remote control of the operating mode (RUN or STOP) from the PC/PG possible.

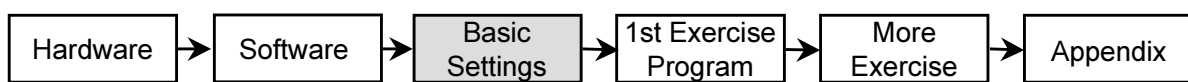


2. From the PC, switch the S7-200 to STOP mode and back to RUN.



The green RUN LED on the PLC lights up in RUN mode. The yellow STOP LED on the PLC lights up in STOP mode. If you can switch the PLC operating modes from the PC, the connection between the PC and the PLC is configured correctly.

If the mode does not change, check the connecting cable between the PC and PLC and the baud rate setting on the PC/PPI cable, and ensure that the correct COM port is selected in the menu **Setup > Communications**.

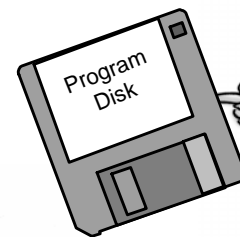


# Writing Your First Program

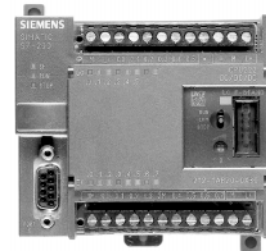
Great, the first functionality test has been successfully performed. The controller is up and running and the data transmission to the PLC works. But how do I write a program?



Now you will quickly learn how to program the basic functions of the PLC in small steps using the small exercise program on the diskette supplied as a basis.



First of all, you will learn how to open an exercise program from the diskette (or hard disk) and transfer it to the PLC with the STEP 7-Micro/WIN programming tool.

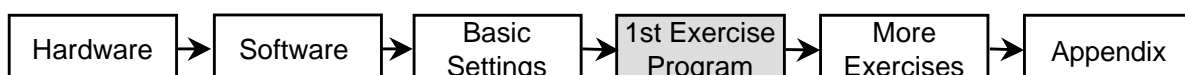


After that, you will familiarize yourself with the function of the program you have transferred and test it.

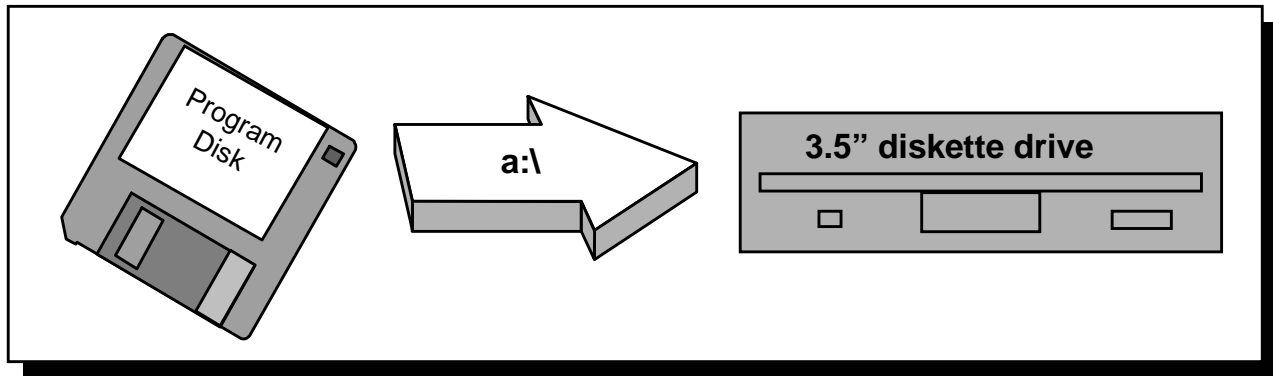
With a little basic knowledge about "logic operations" you will be able to analyze the elements of exercise program 1 and understand the statements in it.



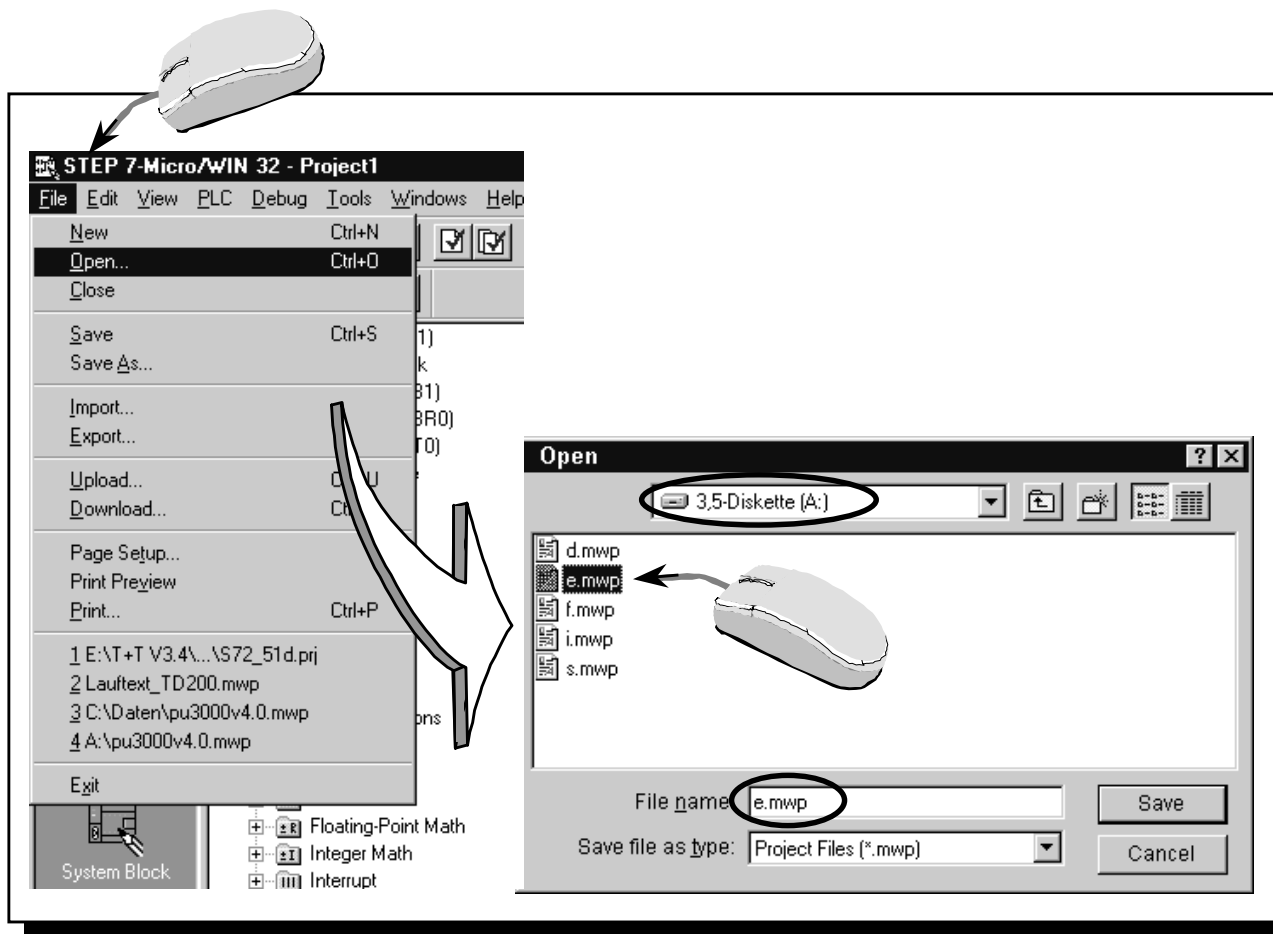
In the Section "More Exercises" you will learn the most important knowledge you require to write your own programs by changing the exercise program.



# Opening the 1st Exercise Program



1. Insert the exercise diskette from the starter kit in your 3.5" diskette drive. The diskette is in the envelope on the last page of the manual.

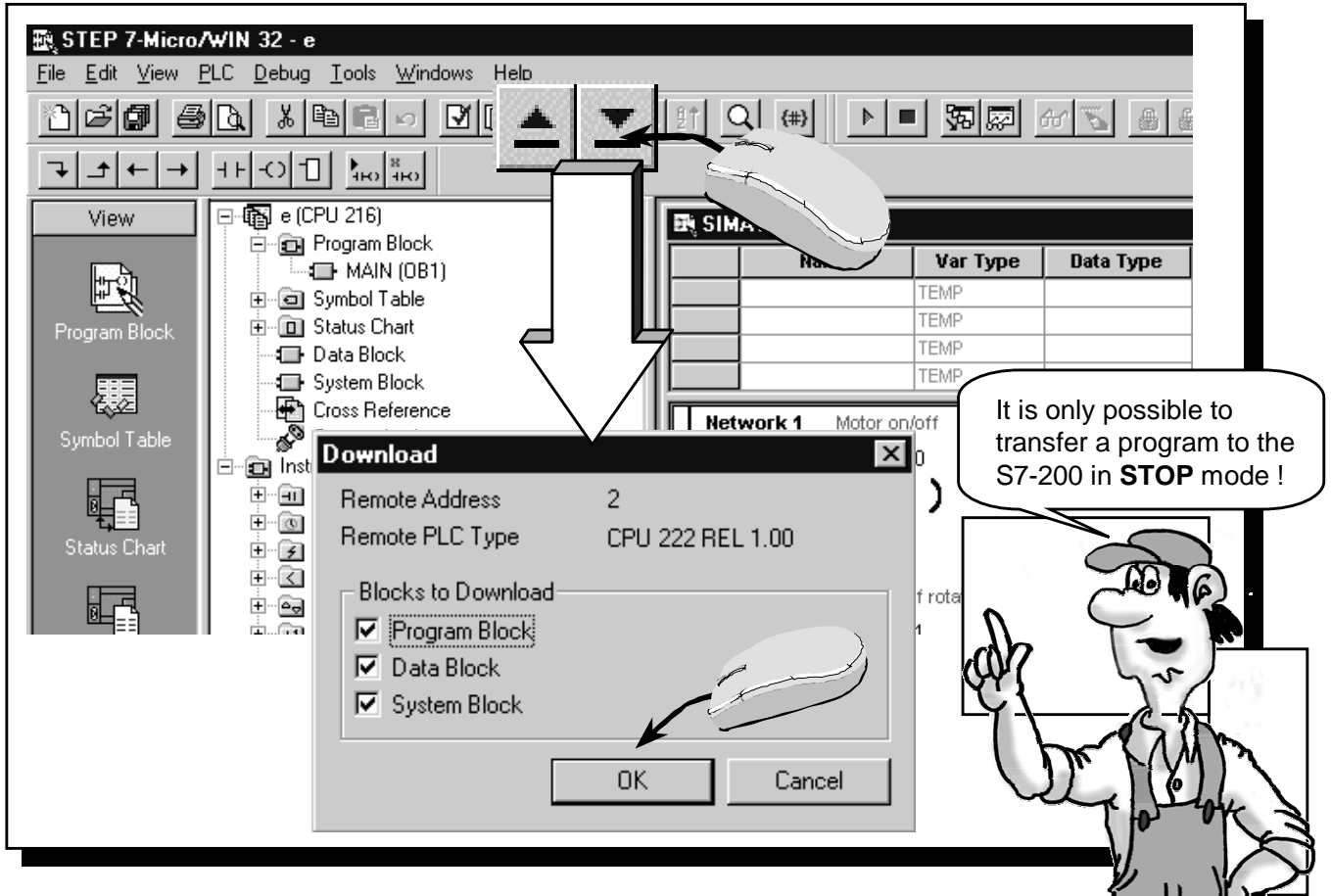


2. From the menu item **Project > Open** load the 1st exercise program from the diskette into the STEP 7-Micro/WIN Editor. The letter of the program name indicates the language of the comments in the program (for English pick E.PRJ). You will need to select your 3.5" diskette drive first.

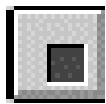


The STEP 7-Micro/WIN programming tool has now loaded the exercise program into the working memory of the PC/PG and is showing the first steps of the exercise program on the screen.

# Downloading the 1st Exercise Program



It is only possible to transfer a program to the S7-200 in **STOP** mode !

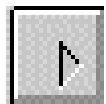


Click this icon with the mouse to switch the PLC to STOP mode when the mode selector on the PLC is set to TERM or RUN and the PLC was in RUN mode (LED STOP lights up).

**Caution:** This action means that a connected machine is no longer controlled.



Click this icon with the mouse to download the (open) program, which is displayed on the monitor, onto the PLC.



Click this icon with the mouse to switch the PLC to RUN mode when the mode selector on the PLC is set to TERM or RUN and the PLC was in STOP mode (LED RUN lights up).

**Caution:** A connected machine may start moving immediately, depending on the effect of your program.



## Warning

Danger of injury and material damage.

You can also load a program from the PLC onto your programming device/PC.



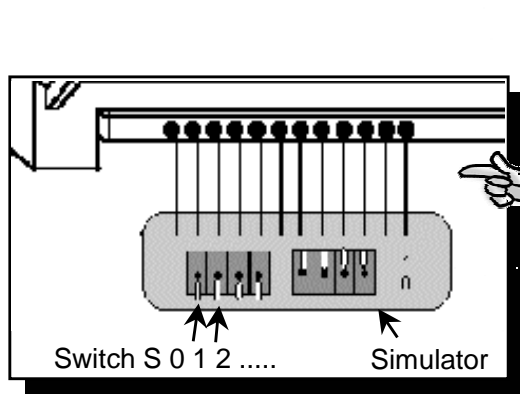
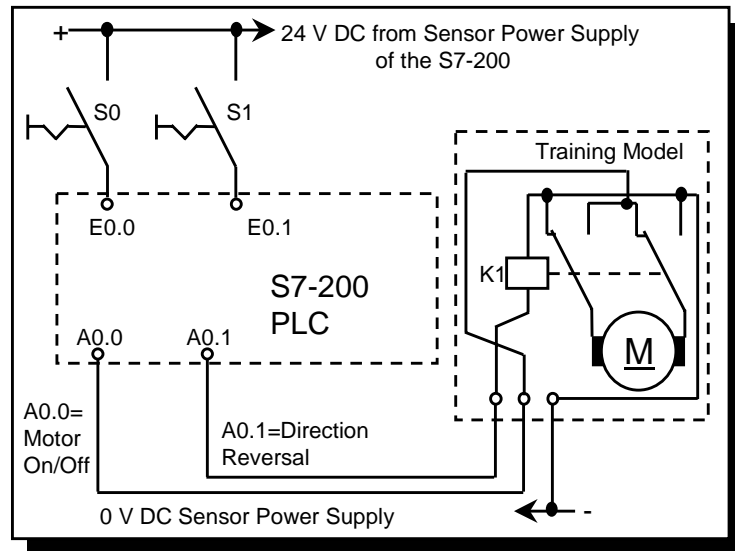
Click this icon with the mouse to upload the program on the PLC onto the programming device/PC. It overwrites the program currently displayed on the monitor. You should therefore always make sure that you always have a current version of your program on the hard disk or diskette when you leave a plant.

# Function and Test of the 1st Exercise Program

In **exercise program 1**, switch S0 is to be used in order to switch on a motor. Switch S1 is used to control the direction of rotation of the motor.

In the **exercise assembly** switches S0 and S1 are on the simulator. The simulator switches 24 V DC to inputs E0.0 and E0.1. The training model is connected to outputs A0.0 (Motor on/off) and A0.1 (direction reversal) of the PLC.

The signal state of input E0.0 is assigned to output A0.0 with the **Program**. The signal state of input E0.1 is assigned to output A0.1.



Now let's test it!  
The operating voltage is connected. The assembly is correctly wired. You have already loaded the program from the diskette into the programming tool and transmitted it from there to the PLC. The PLC is in RUN mode. (The green RUN LED is lit). Now operate switches S0 and S1 and observe the function.

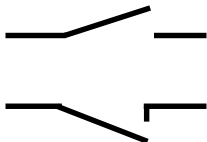

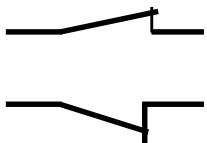

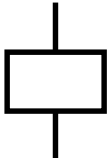
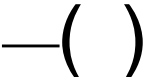
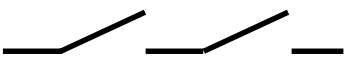
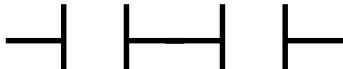
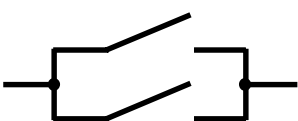
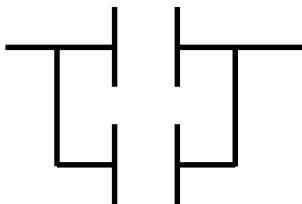
Actions	Reactions		
Switch S0 operated	LED E0.0 lights up	LED A0.0 lights up	Motor spins
Switch S0 & S1 operated	LEDs E0.0 & E0.1 light up	LED's A0.0 & A0.1 light up	Motor spins in the opposite direction



LEDs E0.0 to E0.7 indicate the signal states of inputs E0.0 to E0.7.  
LEDs A0.0 to A0.5 indicate the signal states of outputs A0.0 to A0.5.  
I and Q are the customary international symbols for inputs and outputs.

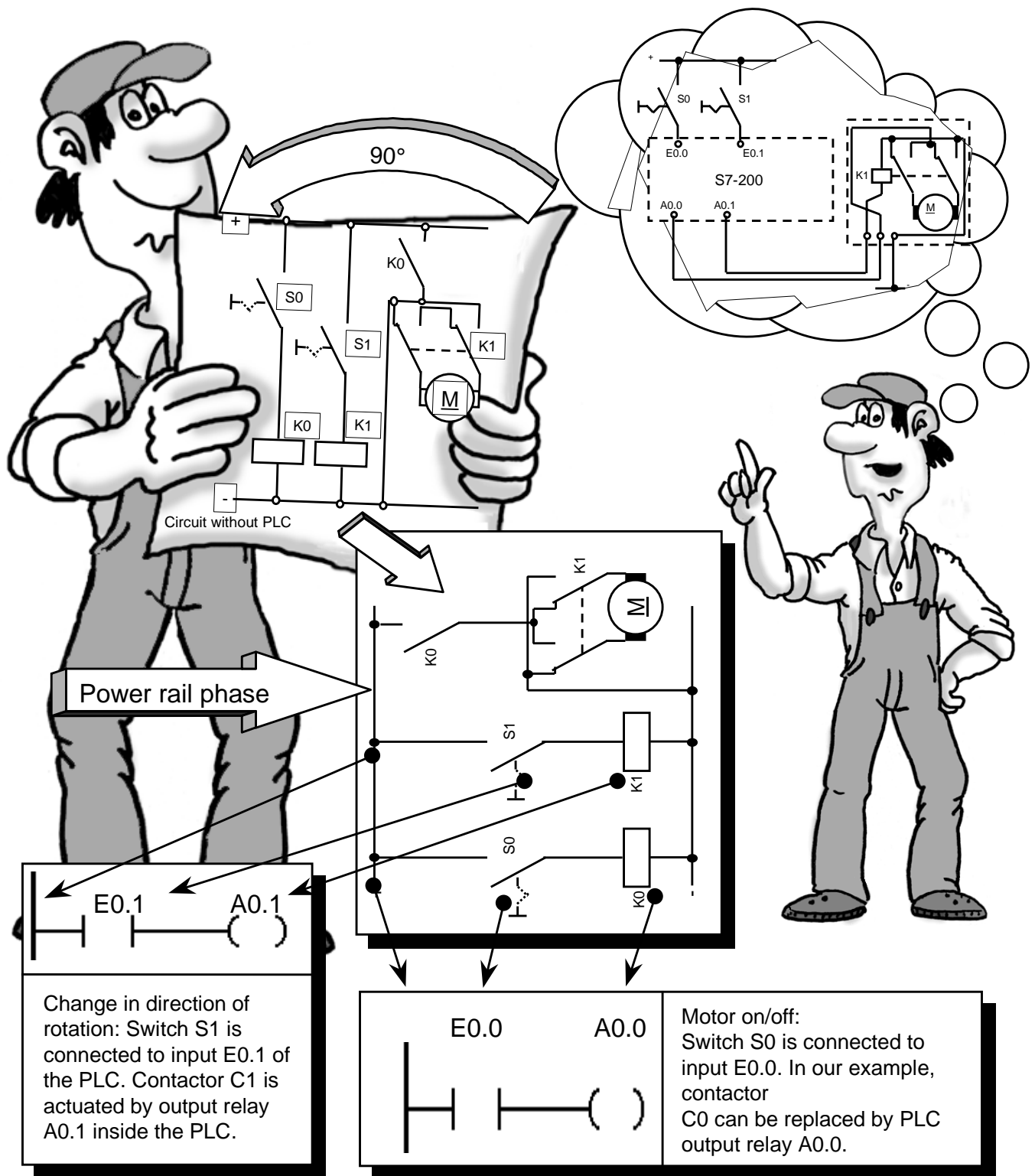


# Ladder Logic Elements

Contactors	Instruction on the PLC/ corresponding function	
	Scan: Is current flowing ? If yes, then the result of the scan is true. (Result is "1")	
	Scan: Is <u>no</u> current flowing ? If yes (no current), then the result of the scan is true. (Result is "0")	
	Coil: If the value "true" (current) is passed to a coil it is activated (The coil starts up).	
	Series circuit: (AND logic). The first switch AND the second switch must be closed in order to pass current.	
	Parallel circuit (OR logic). The first switch OR the second switch must be closed in order to pass current.	

"0" or "1" are the only states in digital control logic. The "0" state is designated as false and the "1" state is true. This is why we say "scan" is "0" (false) or "1" (true).

# Transforming a Circuit Diagram

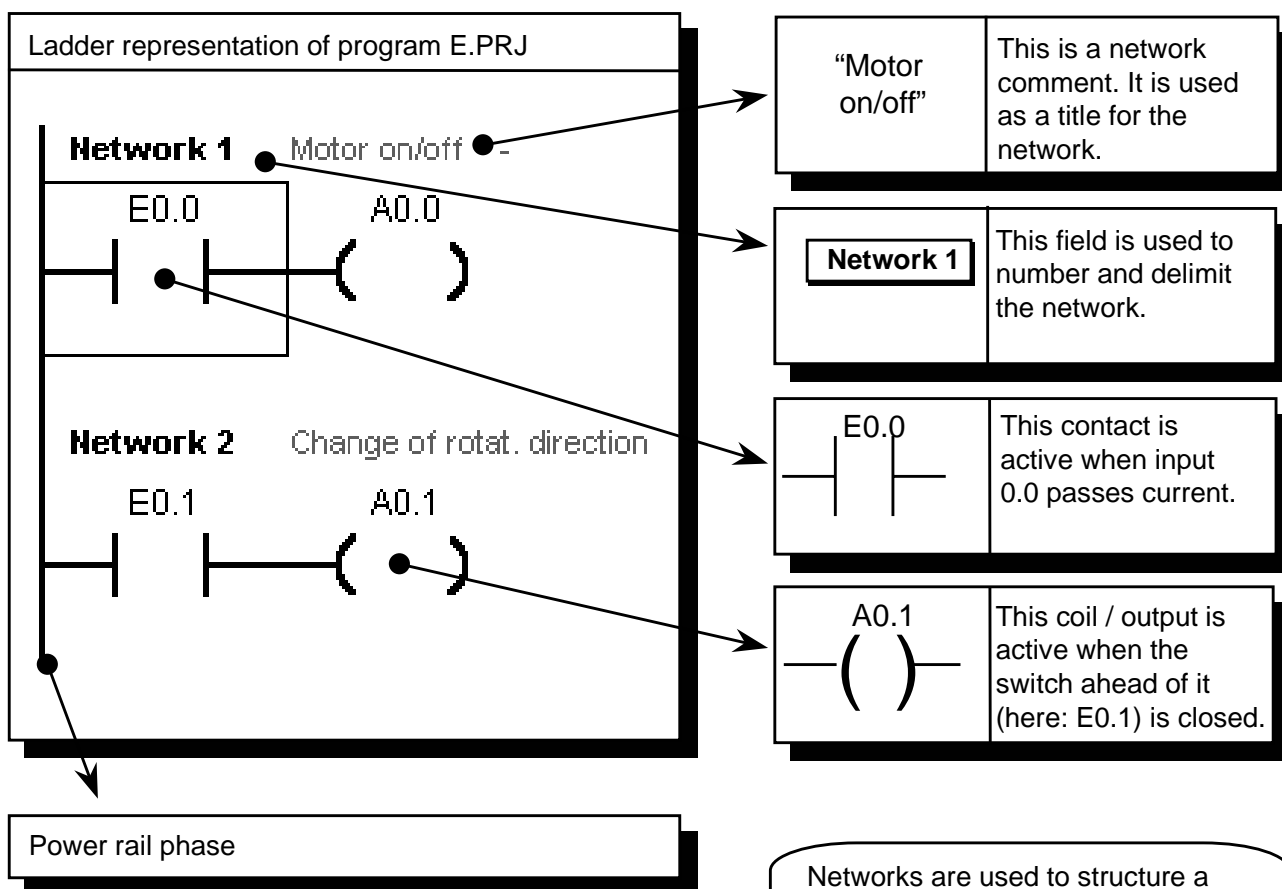


How do you transform a circuit diagram into a PLC program?

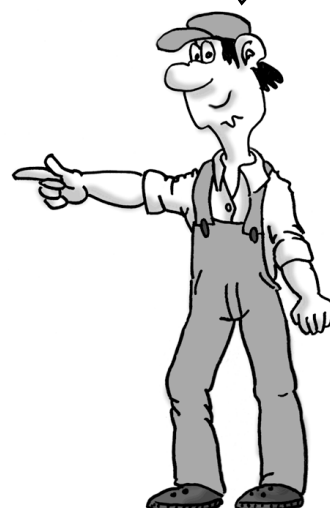
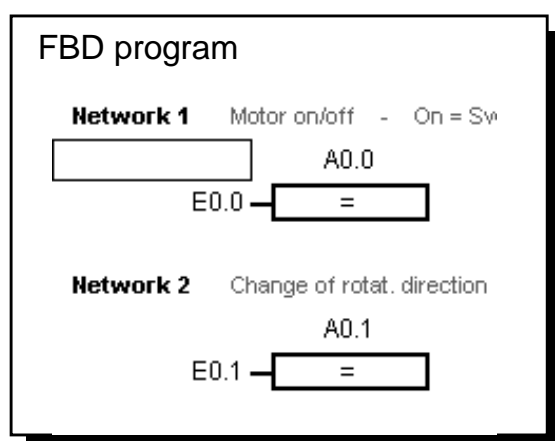
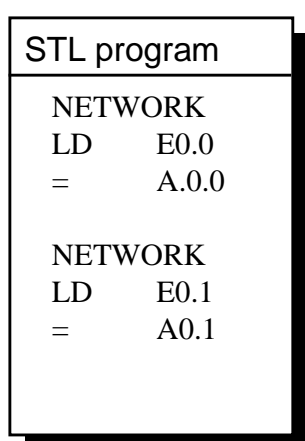
Rotate your circuit diagram 90° to the left. Your power rail will then usually appear on the left, with the grounding rail on the right. In the middle you will see the switching elements of your circuit. The circuit logic of a machine (e.g. time relays or flip-flops) which used to be implemented by wiring together switches, auxiliary contactors and control contactors, etc., is now handled by the PLC. Control elements such as input switches, selector switches, etc. on the input side, and power contactors (such as motor contactors, polarity reversers, valves, etc.) on the output side cannot be replaced by the PLC.

# Elements of Exercise Program 1

Let's have a closer look at the structure of the PLC program in ladder diagram (LAD). This type of representation most closely resembles the circuit diagram.

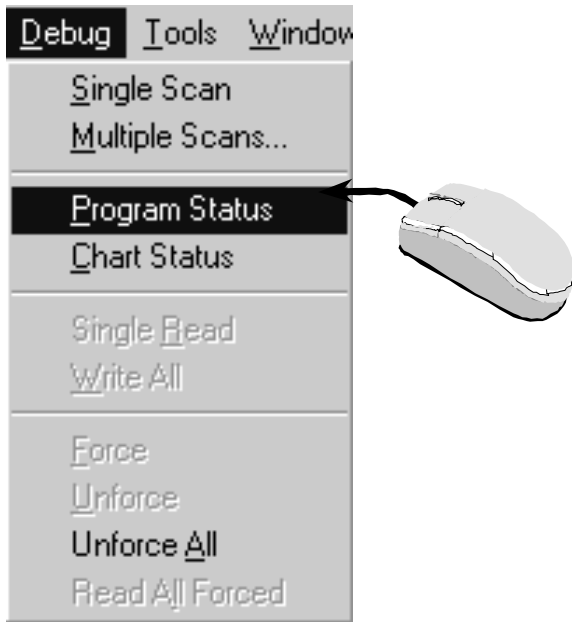


Networks are used to structure a program. You insert each current path into a network.



Statement list and function block diagram, examples of which are shown in the "STL program" and "FBD program" illustration, are further types of representation. You can switch between LAD, STL and FBD in the **View** menu.

# Status View (online)



Select the menu item **Debug > Ladder Status** to activate the status view of the ladder diagram type of representation.

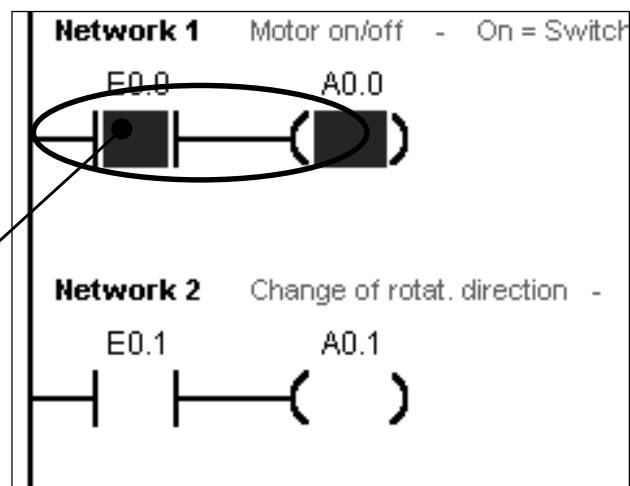
You can now view the status of the operands in the PLC.

In the example, switch S0 is connected to input 0.0.

If you now actuate the switch and observe the program in ladder status view, the operations which are passing current, i.e. are true, are filled in gray.

The status of the operations is read out cyclically from the PLC and updated on the screen.

In this context, online means looking inside the controller with a PC/ programming device in order to view the current states and their cyclic changes.



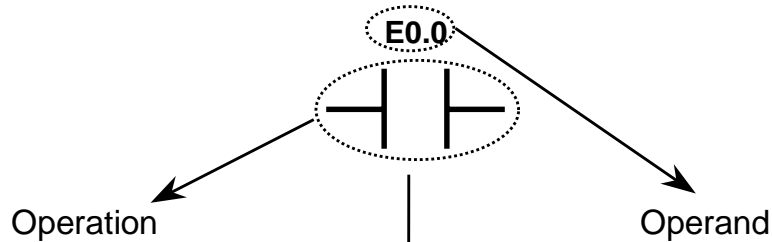
Very fast events cannot be tracked in this way, because the transfer time required to output the data on the screen is too long.

In the representation type function block diagram (FBD) a status view is possible, too.

# Statements

## Control Statements

The control statement is the smallest unit of a PLC user program. A statement consists of an operation part and an operand part.



Operation

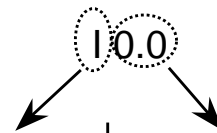
Operand

(Do what ?)

(... to what ?)

The operation part of a statement (in this case an AND logic operation) determines the function that is performed when the control statement is executed.

The operand part of a statement (in this case input 0.0) contains the additional information for the control statement. It consists of an operand identifier and a parameter.



Operand identifier

Parameter

The operand identifier specifies an area of the PLC. In the above example, an operation is performed on an input. Other areas include outputs and bit memories.

The parameter is the address of the operand. It consists of, e.g., a byte and bit address.

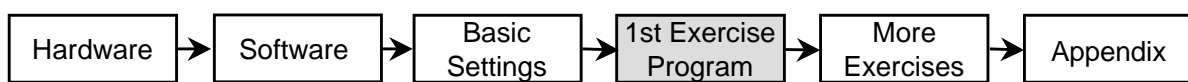
## Structure of an operand

I 0.2

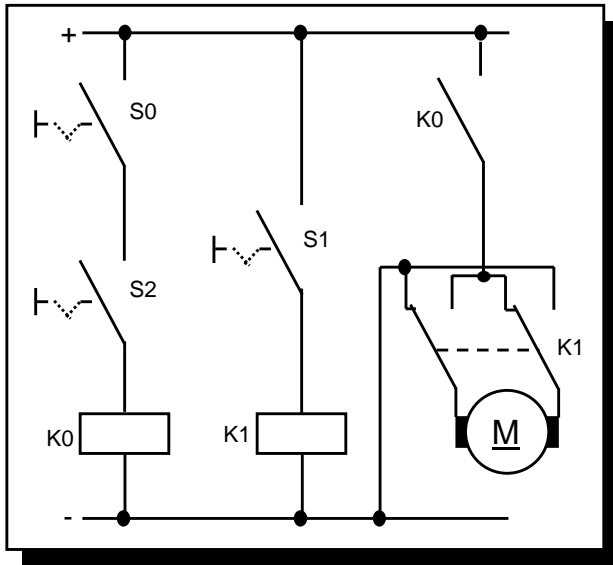
Bit address:  
Decimal point:  
Byte address:  
Operand identifier  
(area identifier)  
Possible areas:

bit number in the byte (0 to 7)  
separates byte address & bit number  
Number of a group of 8 bits

inputs, outputs,  
Internal bit memories, special  
memories, variable memories



# 1. Program Modification: AND Logic Operation



Circuit without PLC

Objective:

Switch S2 is to be actuated in addition to switch S0 in the exercise program, in order to switch on the fan motor.

As before, switch S1 will be used to reverse the direction of rotation of the fan motor.

A verbal description of the function shown above:

The motor should run when S0 AND S2 are actuated. In ladder diagram that means: when contacts E0.0 AND E0.2 are closed, current flows from the power rail to coil A0.0.

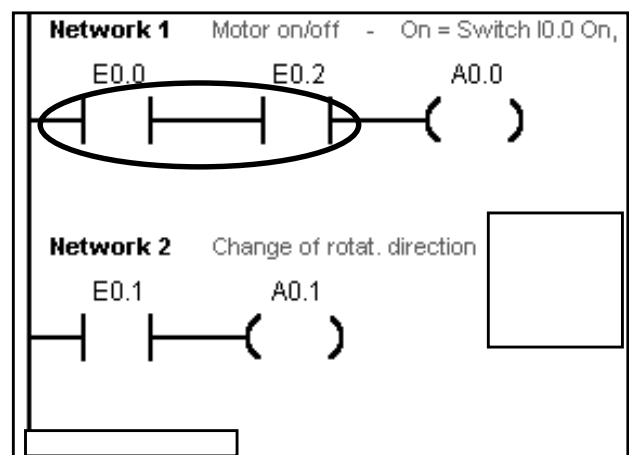
The contacts are connected in series (AND logic).

In our program, the logic is expressed as follows:



We no longer need to wire switch S2, since it is already connected to input E0.2 via the simulator.

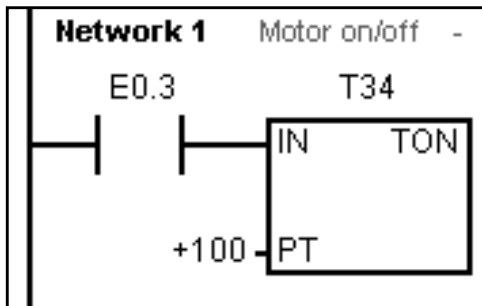
The next page describes how to select, insert, and delete a logic gate, and how to name the operands.



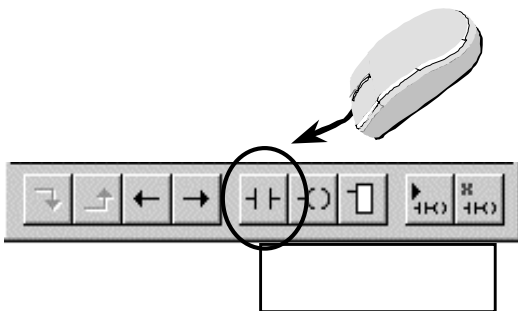
Ladder diagram of the circuit

# Inserting a Logic Gate

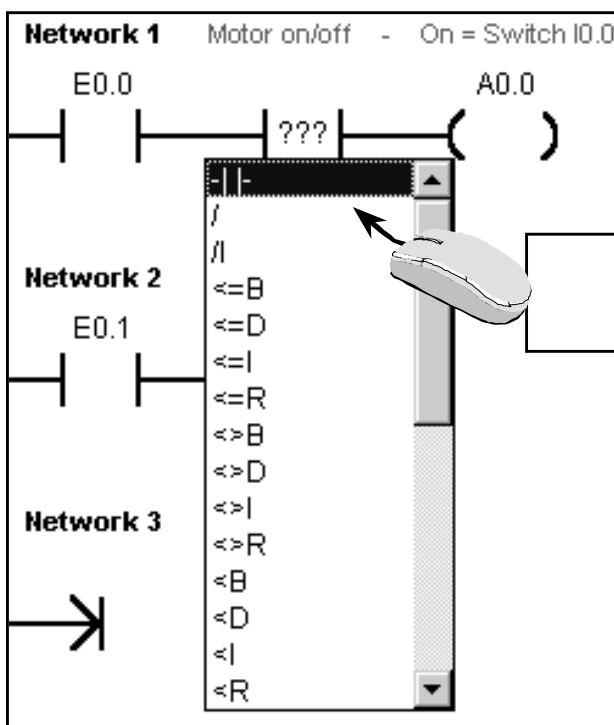
If you want to connect a further normally-open contact for input E0.2 in series between normally-open contact E0.0 and coil A0.0 (i.e. insert an AND link between E0.0 and E0.2), you first need a free location.



Mark the position in the ladder diagram located just before where you want to insert the function.



Select the function in the tool bar or activate the corresponding function key.



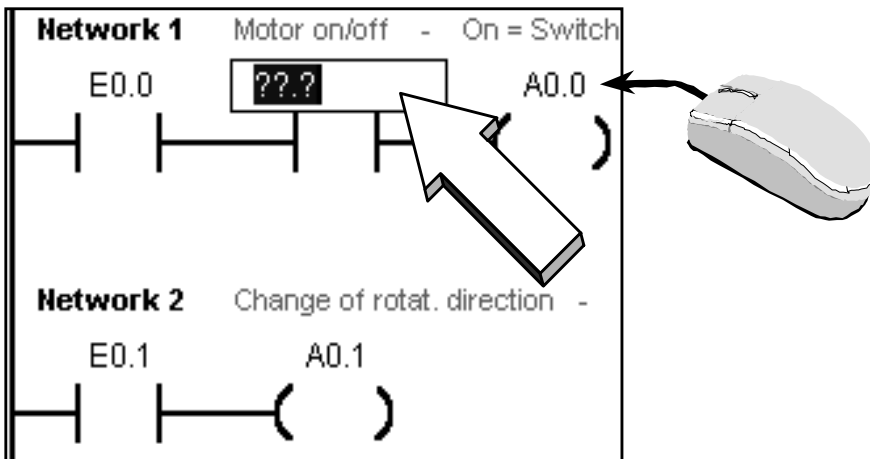
Select the type of function in the menu, in this case a normally-open contact.



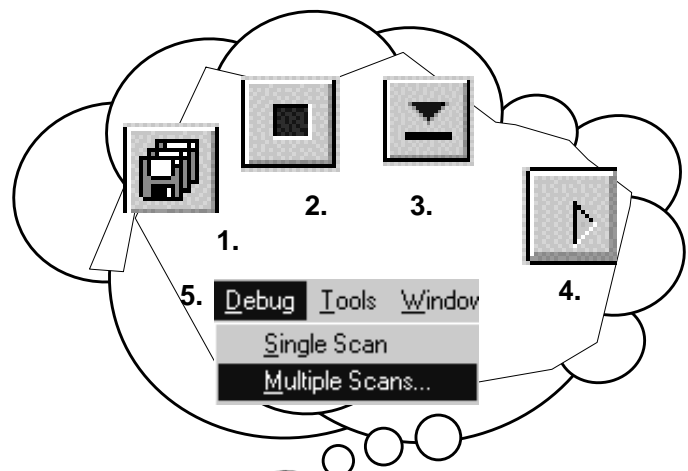
# Entering the Operand and Testing

When the new element has been inserted, you just need to specify the correct operand.  
Click with the mouse into the operand field and input the operand.  
In our case this is: E0.2.  
Confirm the input with the Enter key.

Don't forget to save your changes!



If you want to enter or change the operand later, you can select it again at any time by clicking the field with the mouse.



## If you want to test a modification ...

Load the modifications onto the PLC and test the program. It should be necessary to actuate switches S0 and S2 in order to start the motor.

View your program in ladder status mode and observe the switch settings.

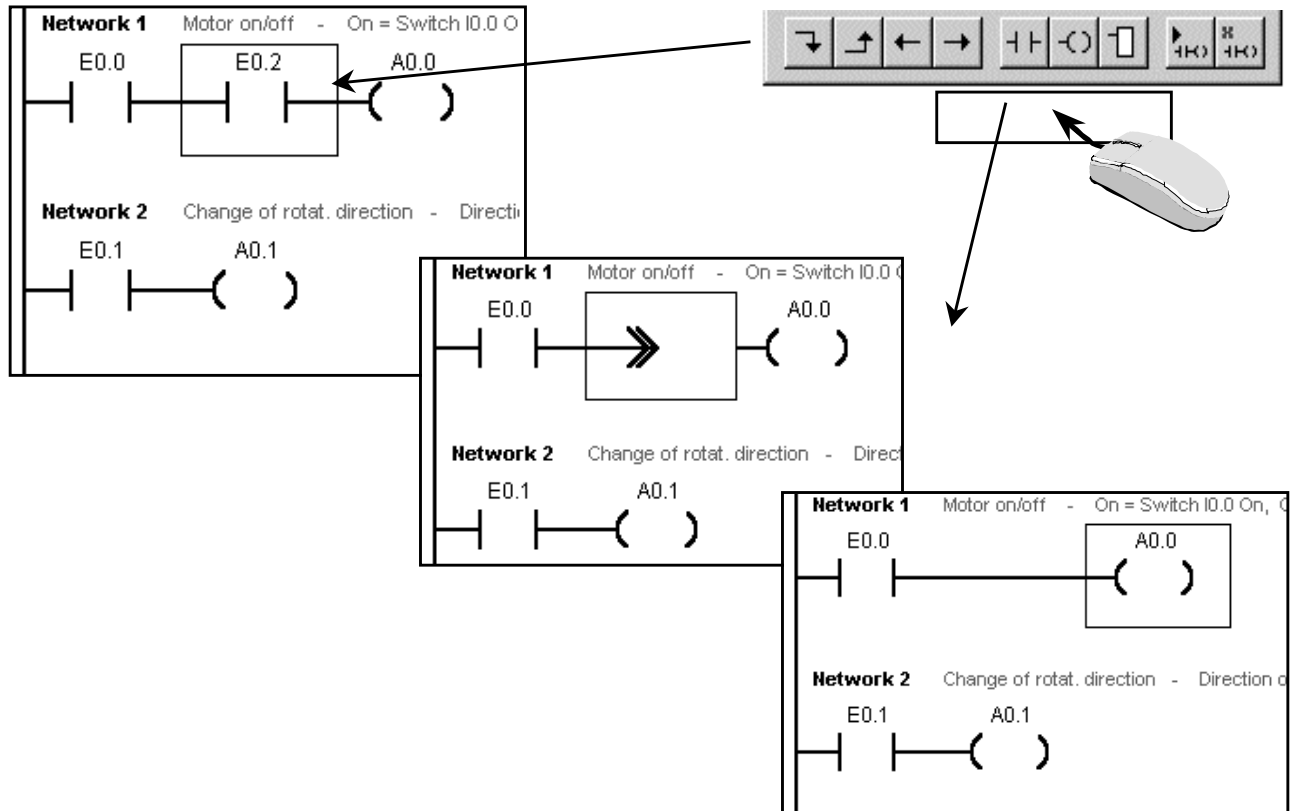


# Deleting ...

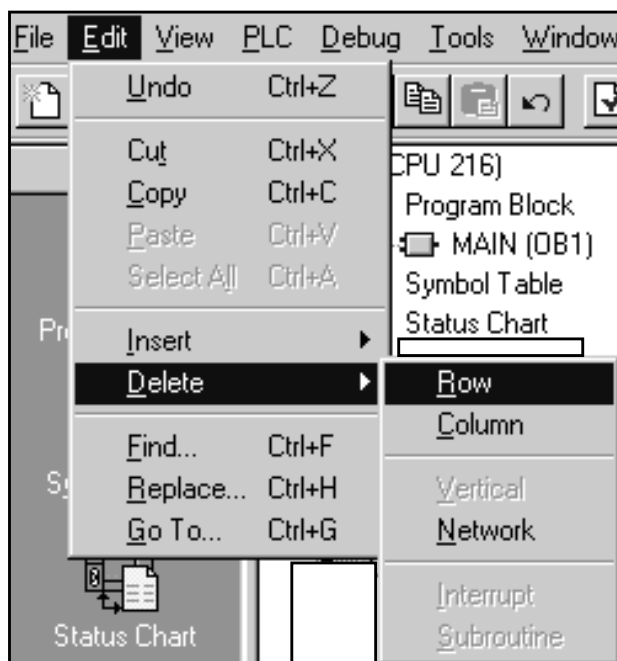
## Logic gates

If you want to delete a logic gate again, select the corresponding gate with the mouse and press the **DEL** key.

To close the current path again, you must establish a connection again.



## Columns, rows, networks and lines

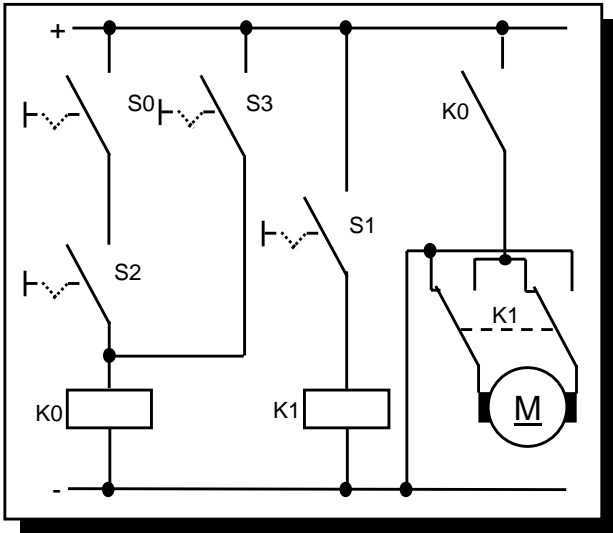


If you want to delete a column, row, line or network proceed as follows:

Select the desired object.

Open the Delete window by selecting **Delete** from the **Edit** menu.

## 2nd Program Modification: OR Logic Operation



Circuit without PLC

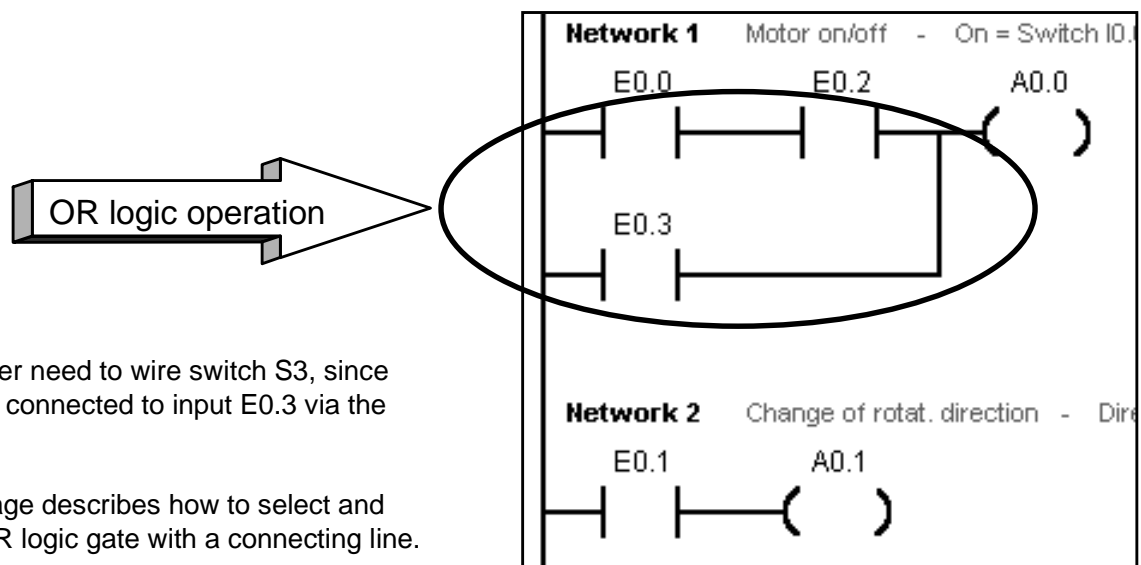
### Objective:

Switches S0 and S2 in the exercise program are to be actuated in order to switch on the motor. Switch S3 alone is to be used as an alternative in order to switch on the motor. Switch S1 is to be used to reverse the direction of rotation of the fan.

A verbal description of the function shown above:

When (S0 AND S2) OR S3 are actuated, the motor should run. In ladder diagram that means : when contacts (I0.0 AND E0.2) OR E0.3 are closed, current flows from the power rail to coil A0.0. This is a parallel connection from S0 and S2 to S3 (OR logic operation).

In our program, the logic is expressed as follows:

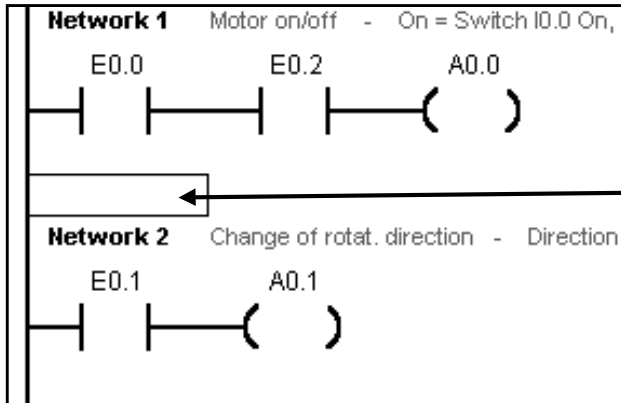


We no longer need to wire switch S3, since it is already connected to input E0.3 via the simulator.

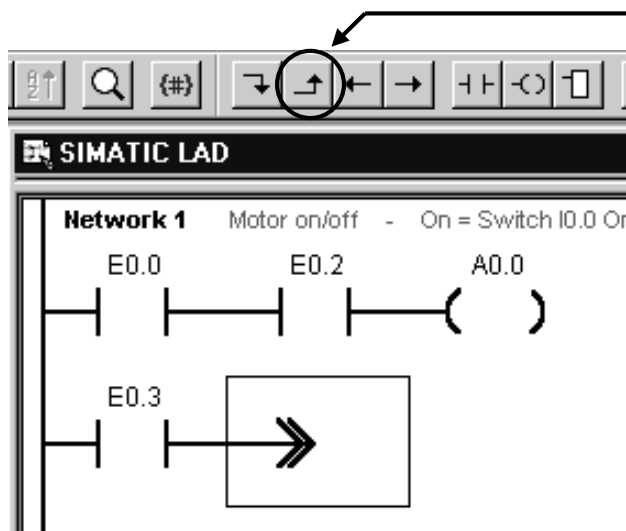
The next page describes how to select and insert an OR logic gate with a connecting line.

Ladder diagram of the circuit

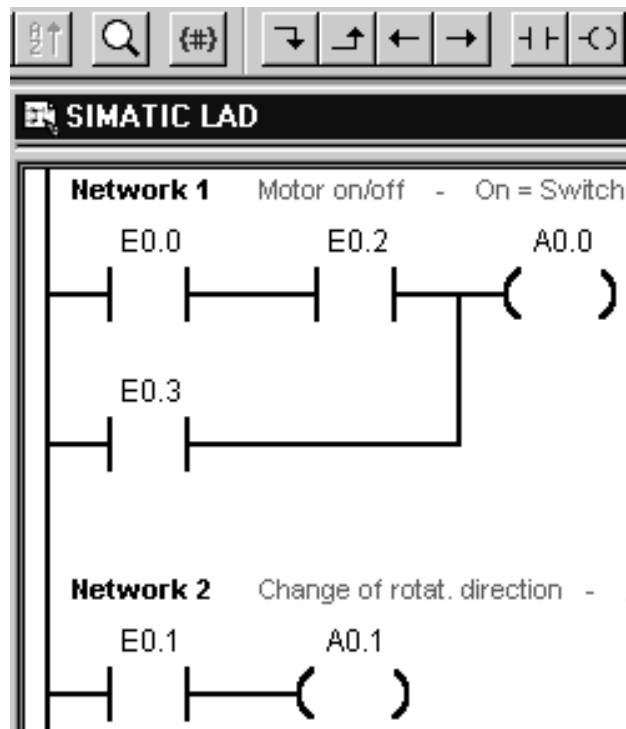
# Inserting an OR Element



1. Select with the mouse a free position in the same network and insert a normally-open contact in the following line.



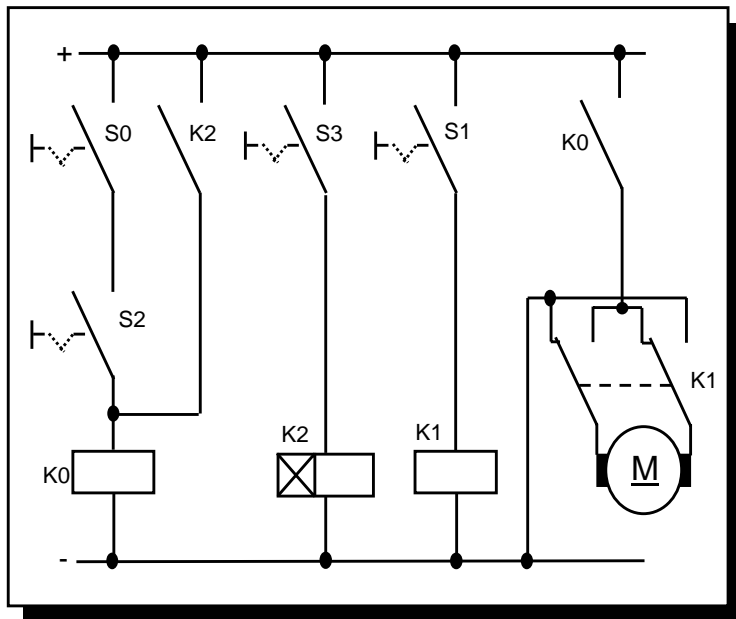
2. Click with the mouse the **Line up** symbol in the LAD tool bar.



3. Now the OR Element is complete.

Your program is now as shown on the previous page. Save your program and load it onto the PLC. Test the function.

### 3. Program Modification: On-Delay Timer



Circuit without PLC

#### Objective:

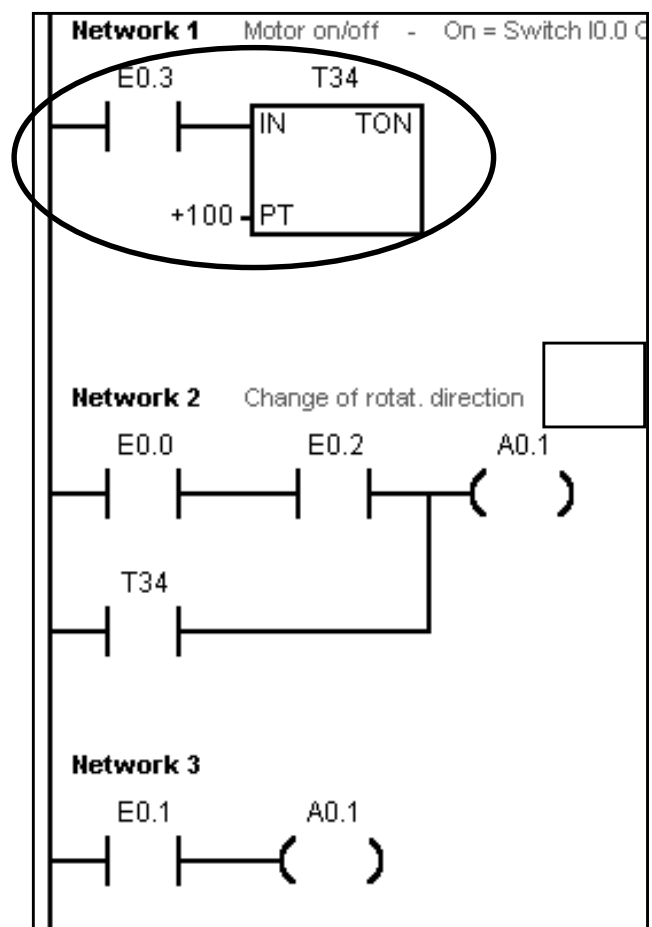
In the following program modification, a time is to be inserted in exercise program 1 as an on-delay.

When input E0.3 (S3) is activated on the simulator, a waiting time is started. Output 0.0, and thus the motor, is not activated until the waiting time has expired.

Wiring work on the PLC because of the additional timer function: NONE.

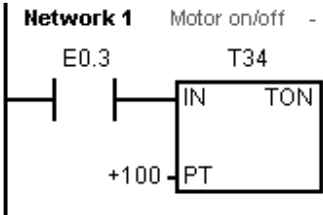
All necessary switches and actuators are already wired. The timer relay is replaced by a PLC function.

The following page introduces the on-delay timer function of the S7-200, which we will then include to obtain the opposite program.



Ladder diagram of the circuit

# The On-Delay Timer Function



You require an on-delay of 1 s. The S7-200 (CPU 221) control has 256 timers. The timers have the designations T0 to T255. Use T34 in this example.

Before the timer will run, it must be started. In our example, this is done with E0.3 at enable input “IN”.



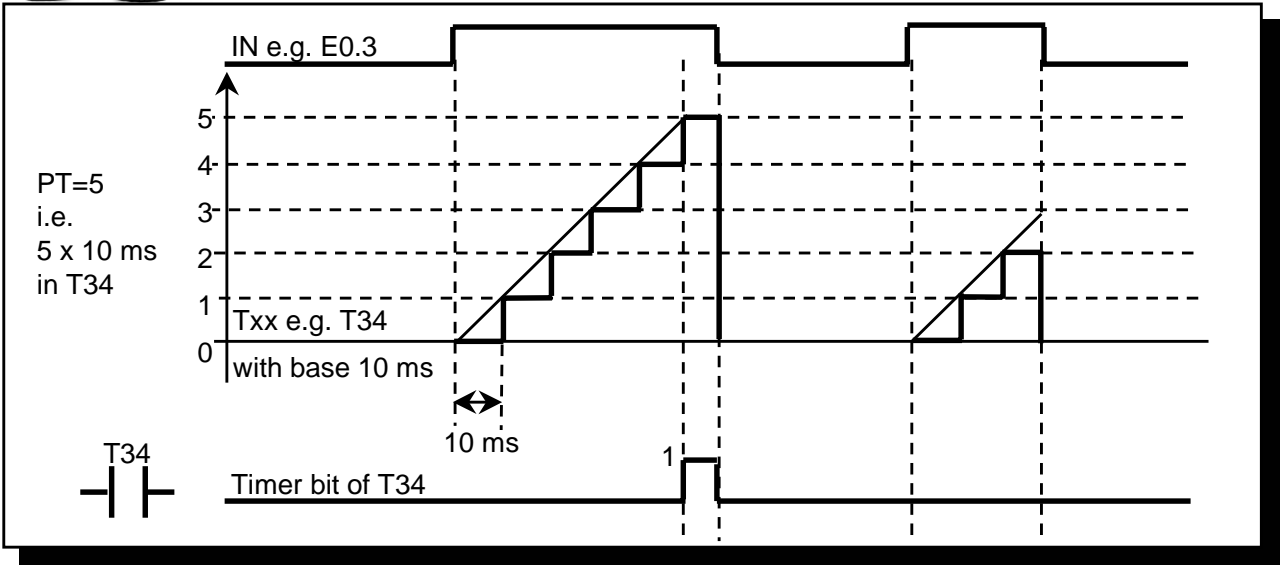
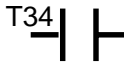
Set the time value of 1s by writing the value 100 to PT. The time value is calculated from the factor PT (here 100) and the timebase of the time. (T34 has a timebase of 10 ms, 100 x 10 ms = 1s).

Timebase	Txx
1 ms	T0, , T32, T64, T96
10 ms	T1-T4, T33-T36, T65-68, T97-T100
100 ms	T5-T31, T37-T63, T69-T95, T101-T255



For CPU 212, timers with timebases of 1 ms, 10 ms and 100 ms are available. Timer must not be put to a dual use.

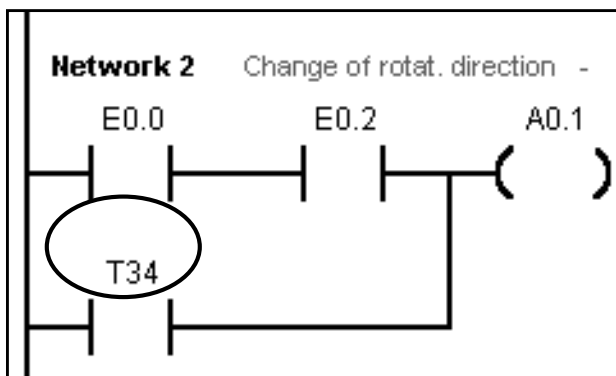
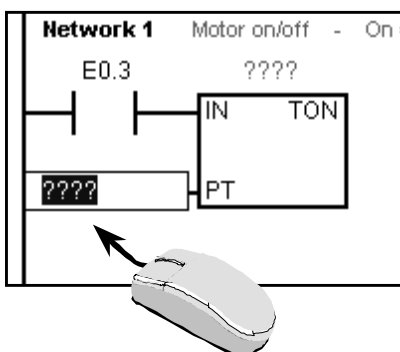
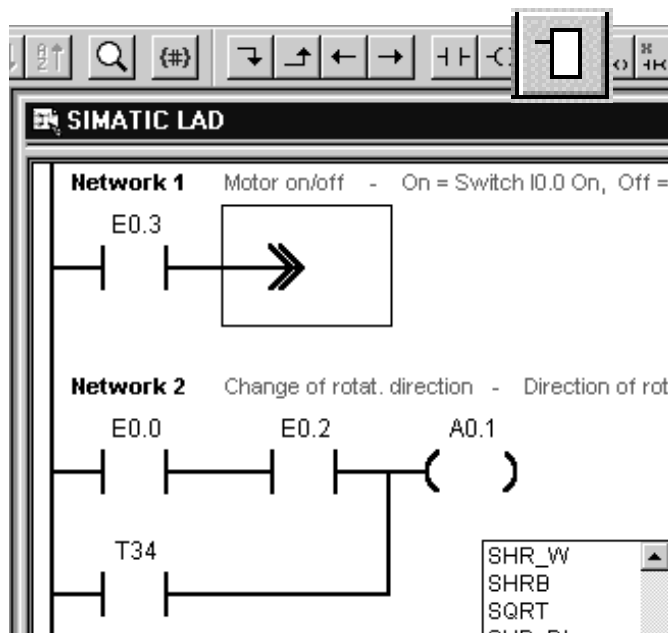
Every timer in the S7-200 has a status bit (time elapsed/not elapsed). This bit is called the timer bit. For the time T34 it is called T34.  
Input E0.3 must have the value “1”, that is 1 s, before timer bit T34 is “1”. If the delay has not elapsed when the enable signal “IN” is canceled, the timer is set to “0” and the timer bit is not set (see the timing chart below). You can use the timer bit like normal operands.



Timing chart for a time value of 50 ms

# Programming the On-Delay Timer

## Using Toolbar



1. To be able to evaluate the timer to be inserted in our OR logic operation, you must program the timer yourself in a previous network.

2. First of all insert a normally-open contact for input E0.3 in the first column of the new network 1. Then input a timer in the second column as the on-delay timer. Select the **Box-F9** tool bar button.

3. Scroll down until the function TON (on-delay timer) appears and select it by clicking the mouse.

4. Select the fields for PT and TON and input your values, in this case 100 and T34.

5. In the last step, the timer bit T34 must now appear in network 2 in the position of E0.3. Make the correction.

Save the program and transfer it to the PLC. Test its function.

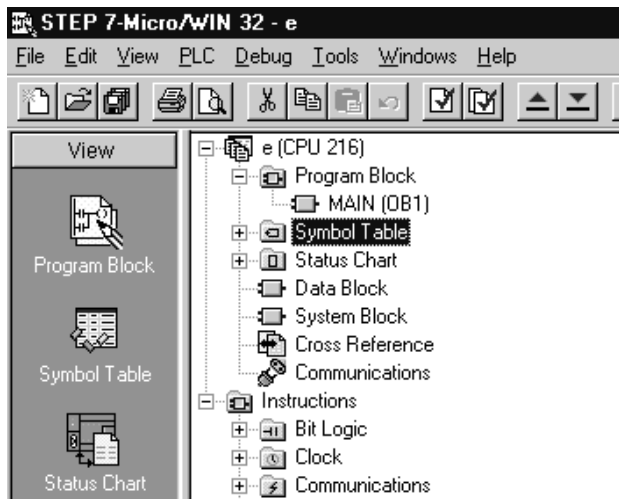


# “Plain Speaking (1)”

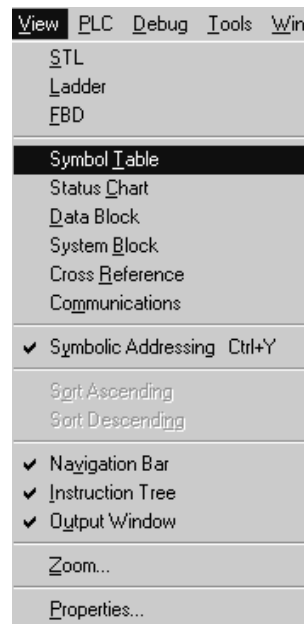
So far you have been working in the PLC program with operands in the “PLC language” such as E0.3 or T34, but if the program is longer, it is not easy to read with these operands.

It would be a good idea if we could have the switch designations or some other plain text.

This is exactly what you can do with symbolic programming.



1. For symbolic addressing, the symbol table must be filled in. Select the symbol table from the operations tree or by selecting **Symbol table** in the **View** menu.



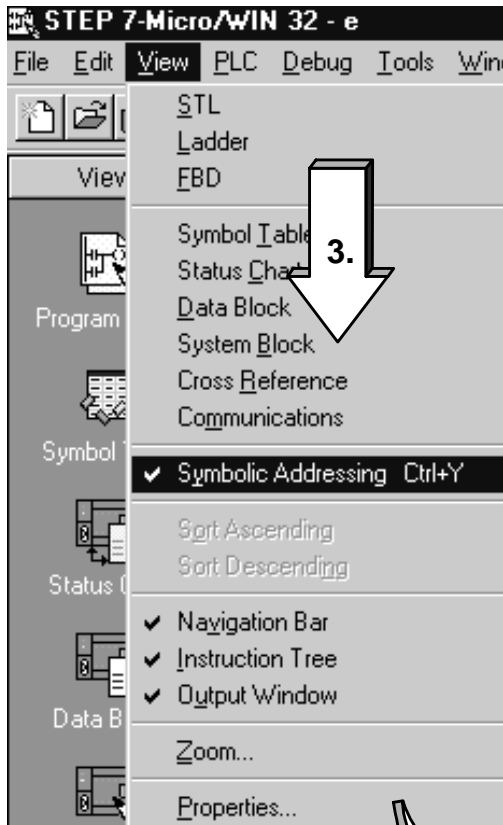
	Name	Address	Comment
1	S0	E0.0	Switch 0
2	S1	E0.1	Switch 1
3	S2	E0.2	Switch 2
4	S3	E0.3	Switch 3
5	Motor_On	A0.0	Turns motor ON
6	Direction	A0.1	Changes the direction of the motor
7	Delay_1s	T34	Waits 100+10ms = 1 second
8			
9			

2. A window is displayed in which you can edit the symbol table. Enter the element that you want to appear as plain text later on under “name”. Under address, enter the operand that you want to have replaced by a symbolic name. Under comment, you can enter a text that you find helpful. Don’t forget to save your work.

Continued on next page

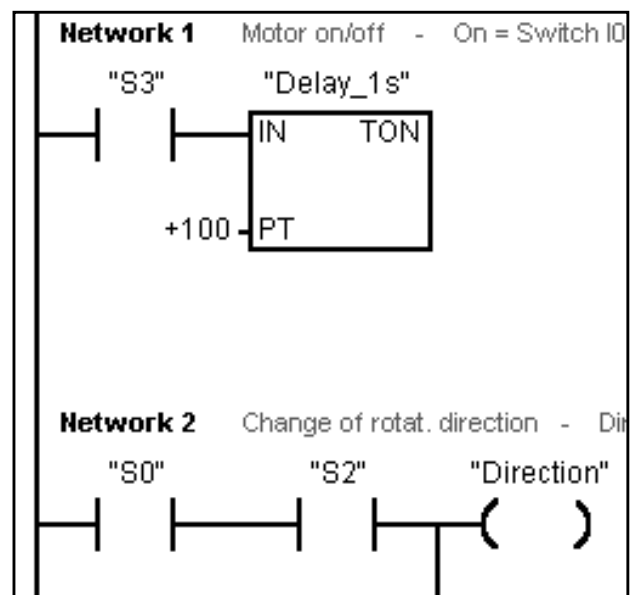


## “Plain Speaking (2)”



3. With the menu item  
**View > Symbolic Addressing**  
you can switch to the display mode for  
symbolic addressing.

4. If you have selected the symbolic  
addressing and changed to LAD, STL or  
FBD, the symbolic addresses should now  
be visible.



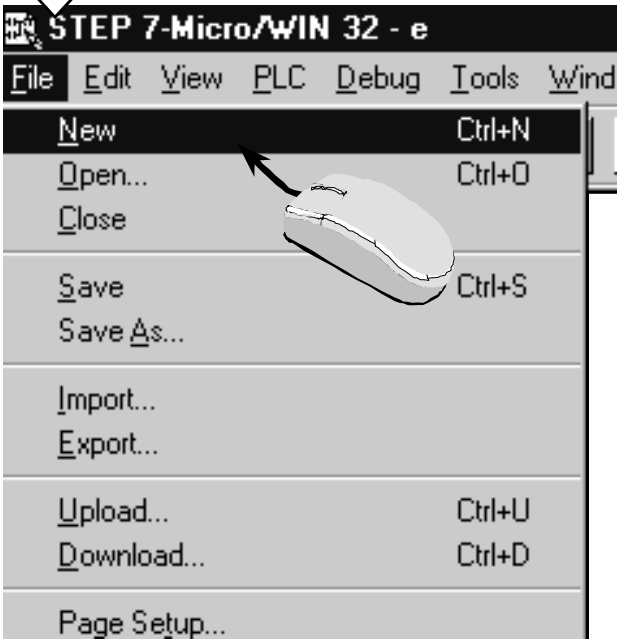
5. The figure above shows the ladder diagram of the user program with symbolic addressing.  
You can, of course, also program with symbolic addresses.  
As the operand, you must then enter “S3”, for example, you only have to make sure  
that this operand has already been entered in the symbol table.

# Creating a New Project (1)

If you want to write a new program of your own, you need a sort of container to put your program file in. In STEP 7-Micro/WIN, this container is the project.

A S7-200 project contains, next to the program file, all additional information to your project, as it is, for example, symbol table, comments etc. .

1.

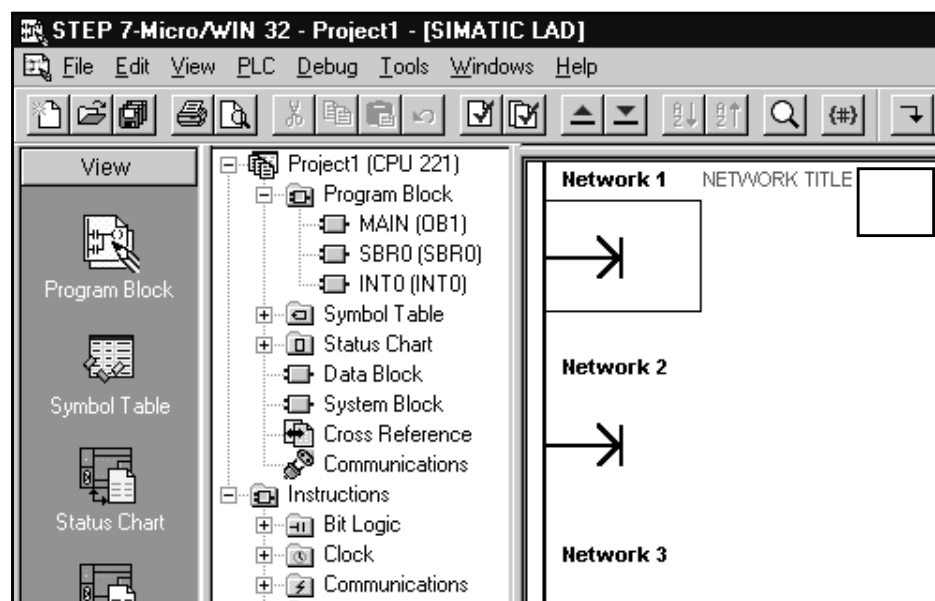


1. Create a new project with menu **File > New** .

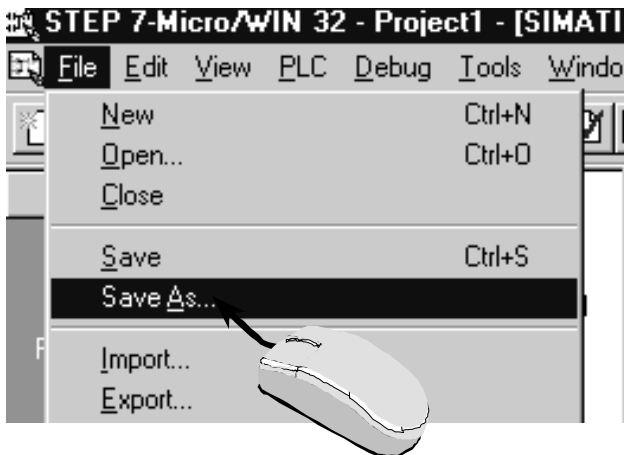
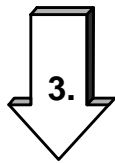
2. The result is a project with the name Project1. You can start programming here.

2.

See the next page for more.

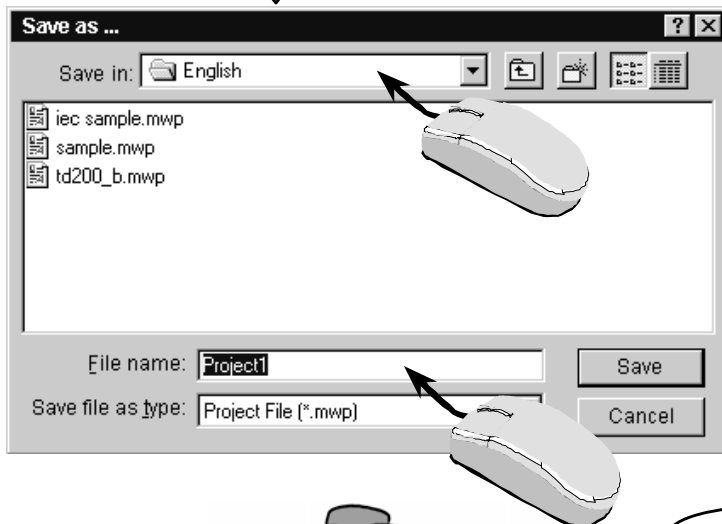
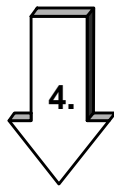


## Creating a New Project (2)



3. Save Project1 immediately or after or during programming under a name that means something to you.

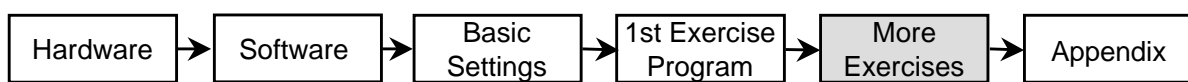
With menu **File** > **Save As...**  
you have the relevant input box displayed.



4. In this box you can give a name to your project and select the hard disk drive and the path/folder where you want to store your project.  
Terminate your input with **Save**.

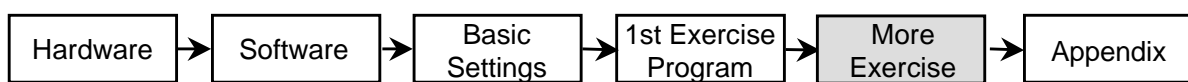


After these steps, you have now created a new project with a new program file. You can now work as in the usual way.





Congratulations, now you know how to program the Micro PLC S7-200.  
If you want to learn more, you will find useful information in the appendix.



## Do you want to learn more ?

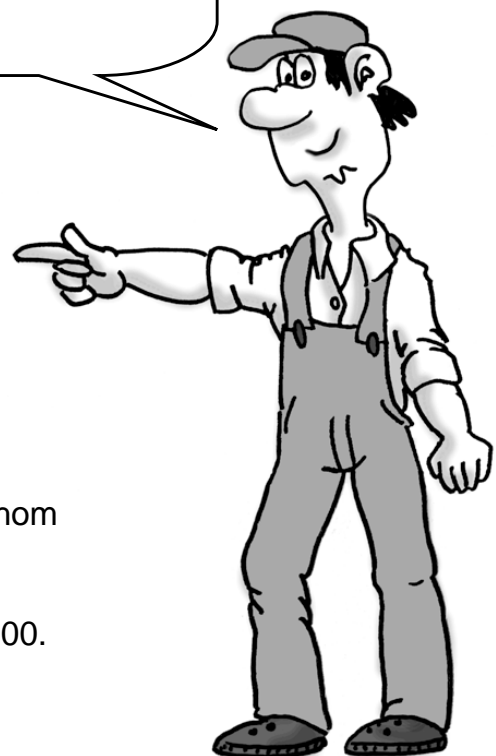
You will find further examples in the “Samples” directory in your STEP 7-Micro/WIN directory.

The examples of the “One Hour Primer” are continued in the “Two Ore Primer”.

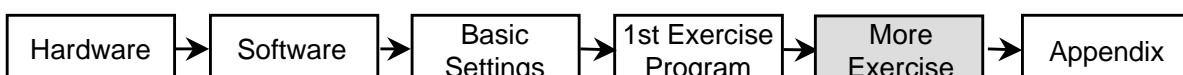
Furthermore a CD ROM with “Tips & Tricks” for the S7-200 is available. You can obtain the “Two Hour Primer” and the “Tips & Tricks” from your SIMATIC representative.

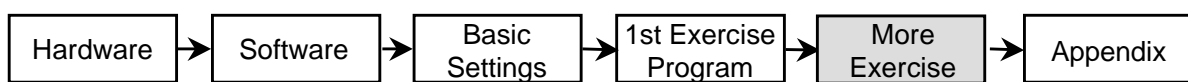
You will find more information in the manuals for the S7-200. For further training you can attend a S7-200 course in your Siemens Training Center or with your SIMATIC representative.

Unanswered questions?  
Technical problems?  
The SIMATIC representatives  
will be glad to help.



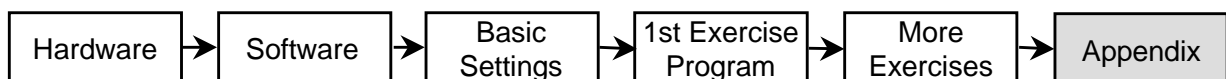
Please contact the SIMATIC representative from whom you purchased the Starter Kit.  
If you cannot contact your SIMATIC representative,  
please call our SIMATIC hotline on +49 911 895-7000.







# Appendix



# Of Bits, Bytes and Words

The smallest unit of information in a digital system is known as a “bit”. A bit can only have the states “0” (i.e. false or untrue) or “1” (i.e. true).

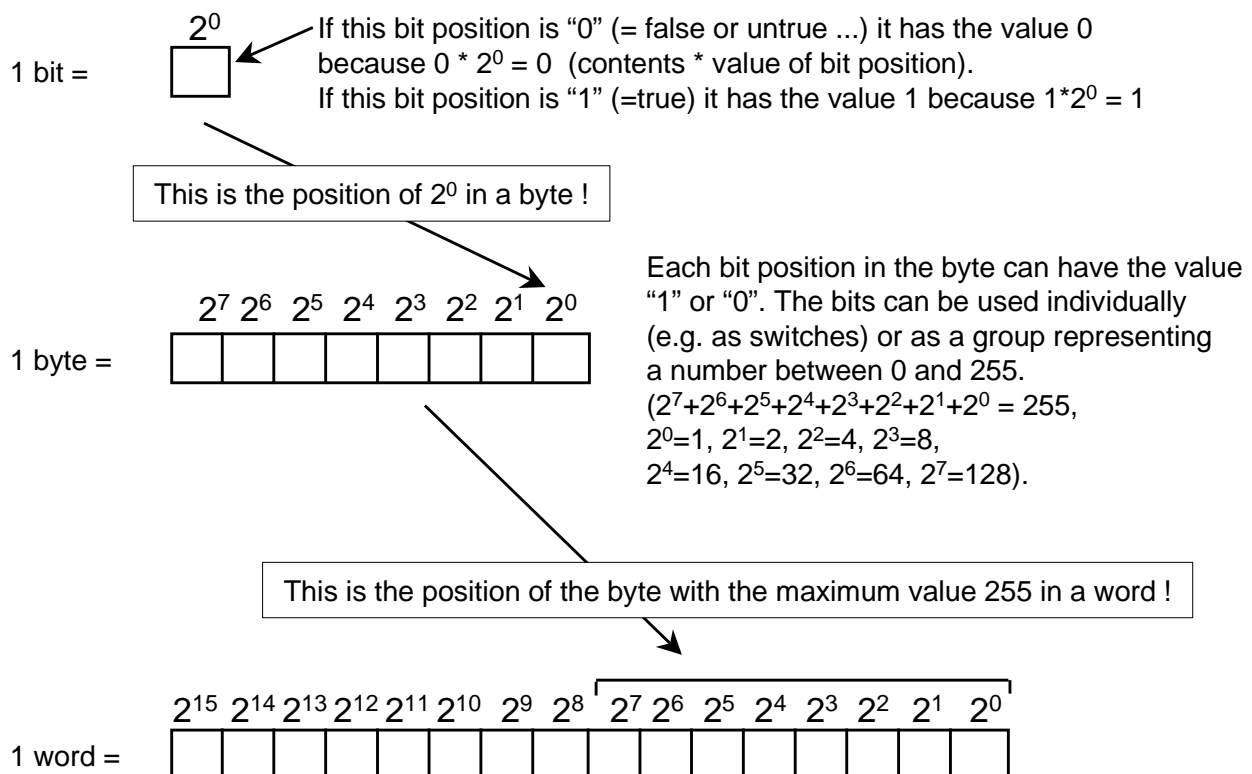
A light switch, for example, only has the states “light on” or “light off”, i.e. the value of the light switch in answer to the question “Is the light on?” is either true (the light is switched on) or false (the light is switched off). The light switch thus has an information width of one bit. The state “light switch on but bulb defective” is ignored in this example.

In a PLC, bits are organized into groups.

A group of 8 bits is called a byte. Each bit in the group is defined exactly by a separate position with its own address. A byte has a byte address and bit addresses 0 to 7.

A group of 2 bytes is called a word.

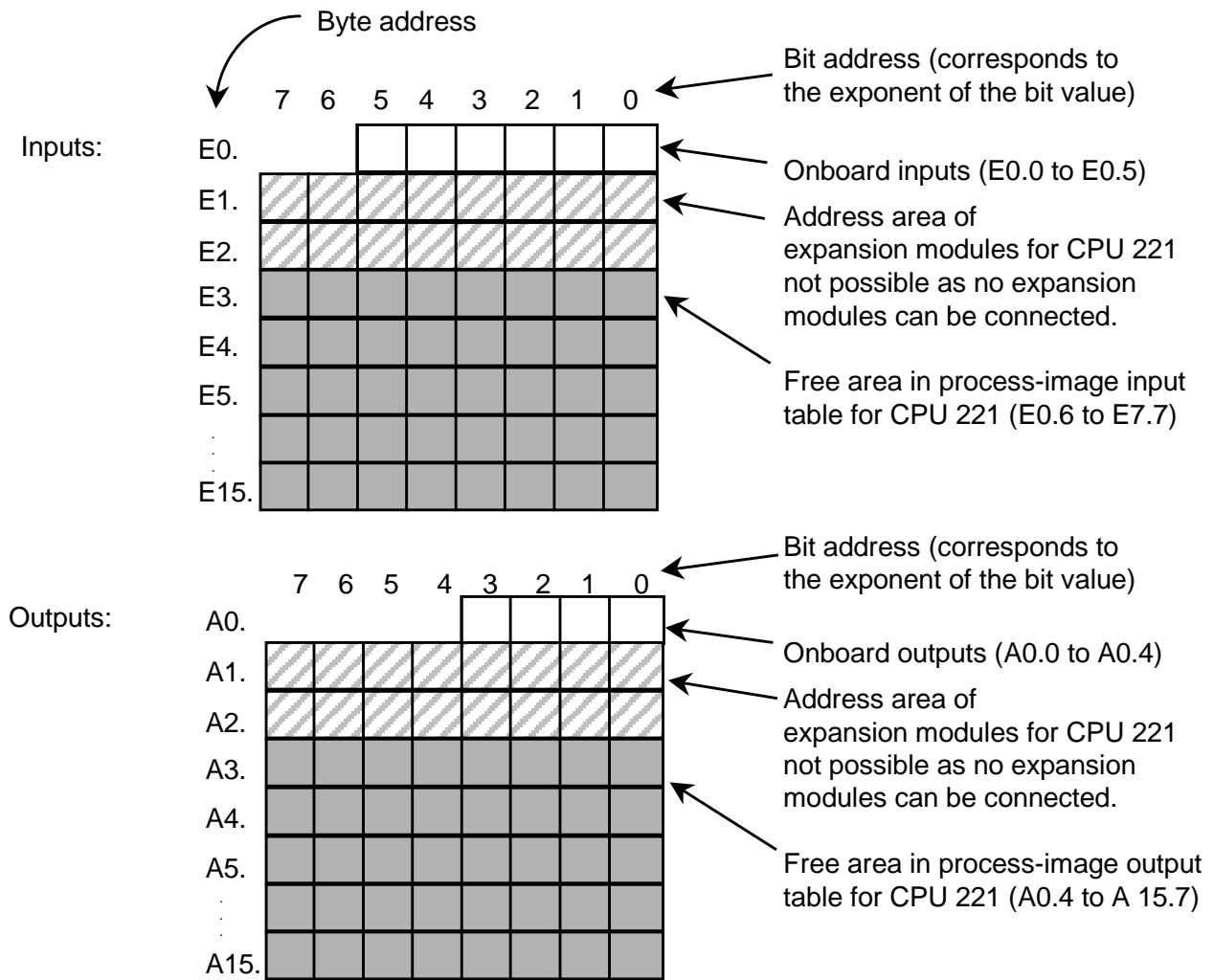
The numbers in this system are binary numbers, i.e they are counted to base 2.



A word in a PLC can represent a number from -32768 to +32767. The bit with the value  $2^{15}$  is used to denote a negative number (when position  $2^{15}$  has the value “1” the number is negative).

# Address Areas of the S7-200

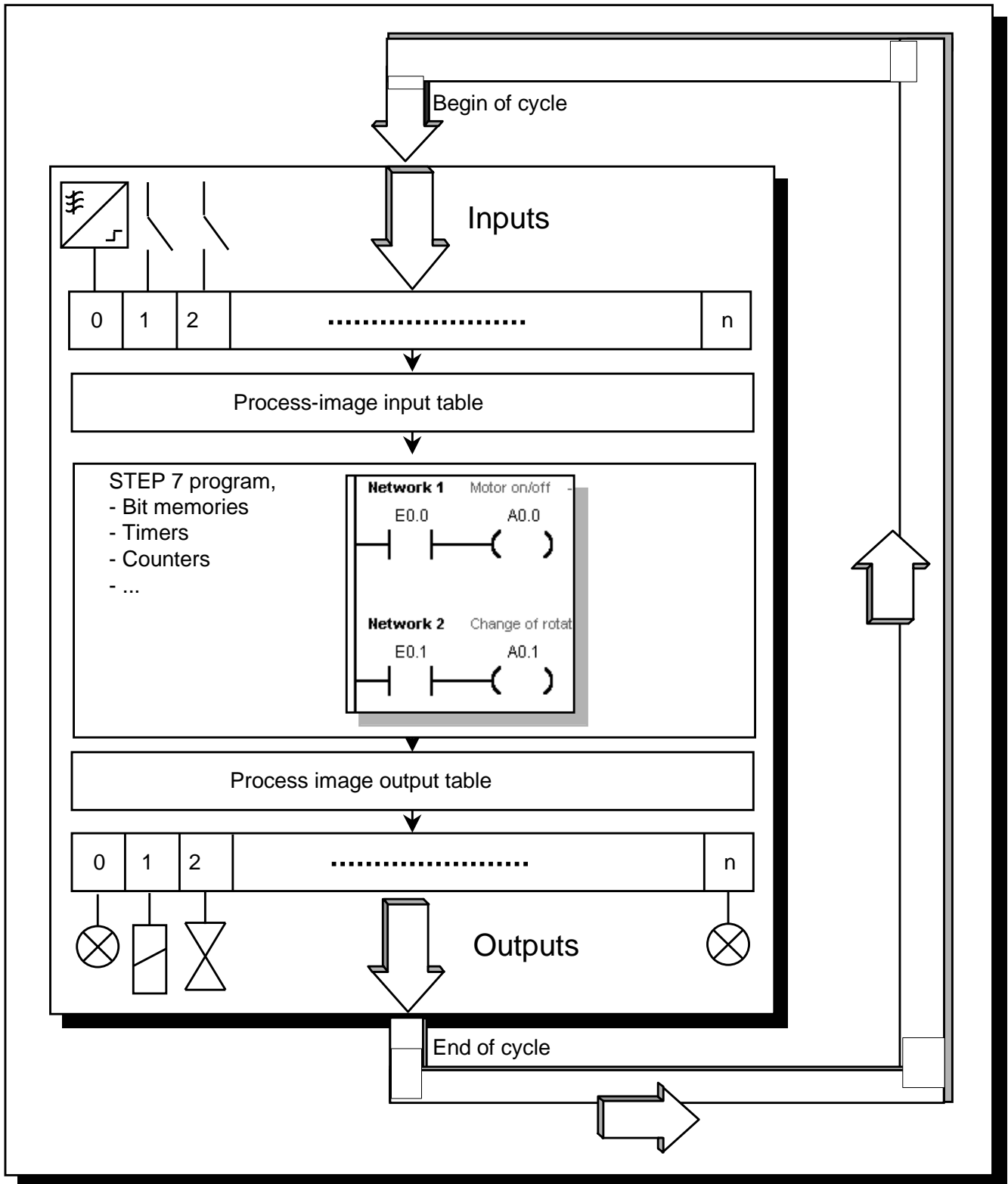
Address areas of an S7-200 PLC with CPU 221



The address areas of a PLC are the memory areas in which the inputs and outputs of the control are mapped. For example, if a voltage is applied to a physical input, this “1” signal is mapped at an address in the memory that is assigned to this input.

The memory areas which form the interface with the terminals (inputs and outputs) of the PLC are called the “process image of the inputs” (PII) and the “process image of the outputs” (PIO). These memory areas contain an image of the signals from the process.

# Cyclic Program Execution in the S7-200 PLC



All SIMATIC PLCs operate cyclically. During each cycle, the switch states are first read in from the inputs, and then stored in the process-image input table. The control program is then executed on the basis of this information.

The outputs in the process-image output table are subsequently controlled in accordance with the circuit logic of the program. The last step is to transfer the states from the process-image output table to the physical outputs. The cycle is then repeated.

To

Siemens AG  
A&D AS MVM  
Gleiwitzer Str. 555

Fax: +49 911 895-2786

90475 Nuremberg  
Federal Republic of Germany

## Response to the "One Hour Primer"

Dear user of the Micro PLC S7-200

We created the One Hour Primer so that, together with the Starter Kit, you can learn to use the Micro PLC S7-200 within a very short time.

We are sure that you will easily find your way into the world of S7-200 with this primer. However, if you do have any suggestions, it is important to us to hear your opinion.

Please send us this form, stating your name and your address so that we can contact you directly.

Thank you

A&D AS MVM

---

## Suggestions, Improvements, Feedback

From

Name	_____	Function	_____
Company	_____	Telephone	_____
Street	_____	City/Postal Code	_____

My suggestions:

---

---

---

---

---

---

---

---