Programm für lebenslanges Lernen / GD Bildung und Kultur — Federal Ministry of Education and Research	Berufsschule für Fertigungstechnik	**Module PLC**	goMOVET.eu

This action based training was developed within the Leonardo Da Vinci Transfer of Innovation Project:

"MODULES FOR VOCATIONAL EDUCATION AND TRAINING FOR COMPETENCES IN EUROPE"
"MOVET"
(PROJECTNUMBER **DE/10/LLP-LdV/TOI/147341)**

Module PLC

The aim of the training is to enable the apprentices to develop the skills, knowledge and competence for competence area 7 of the competence Matrix Mechatronics from the VQTS model (cf. Karin Luomi-Messerer & Jörg Markowitsch, Vienna 2006)

7.3 He/She can integrate and configure program-, control-, and regulation-mechanisms in mechatronic systems, program simple devices (in co-operation with developers) and simulate the program sequence before start-up.

BSFT

Overview of the Module

Competence Matrix „Mechatronics“

Competence area	Steps of competence development					
1. Maintaining and assuring the reliability of mechatronic systems	He/She can perform the basic scheduled maintenance on mechatronic machines and systems and adhere to the equipment maintenance plans.	He/She can master the maintenance procedures for mechatronic systems such as the use of service documents and maintenance plans and, if faced with new challenges, can make the necessary adaptations.	He/She can use preventive maintenance to assure the trouble-free operation of mechatronic systems. In addition, he/she can modify operational sequences to implement quality-assurance measures	He/She can develop the necessary procedures for maintenance of mechatronic devices and systems, and can schedule the maintenance and quality-assurance procedures.		
2. Installing and dismantling mechatronic systems and facilities	He/She can use written instructions to install and dismantle individual components (sensors, actuators, drives, motors, transport systems, racks) that form a functional group of mechatronic systems.	He/She can master the installation and dismantling of mechatronic systems that use several technologies (mechanics, hydraulics, pneumatics, electricalmechanics, electronics), set up the connexion technology, and check the efficiency of the overall system.	He/She can provide independent mechatronic solutions for the construction of production lines, assure their overall ability to function, and, in addition, can use both existing and modifi ed standard components.			
3. Installing and adjusting mechatronic components in systems and production lines	He/She is able to install and adjust standardized mechatronic components, e.g. individual electro-pneumatic valves, sensor and actuator units.	He/She can install and adjust components of mechatronic subsystems (e.g., linear drives, measuring systems, transport systems).	He/She can install and adjust complex mechatronic facilities that include diverse technologies and instrumentation and control (I&C) equipment, adjust the associated parameters, test the facilities overall functions, and assure their reliability			
4. Designing, adapting, and building mechatronic systems and facilities on the basis of client needs and site plans	He/She can use machine tools controlled either manually or via computer-program to fabricate (according to production designs and customer requirements) the individual components for mechatronic systems. He/she can provide simple designs and descriptions of mechatronic subsystems and can use basic CAD applications.	He/She can build simple mechatronic subsystems by using eineering drawing and can install he devices according to specific production needs. He/She can act on extensive knowledge of standards and regulations (e.g. on surface treatments) and is able to use CAD's more advanced functions (e.g. interference check).	He/She can build mechatronic systems by using both original construction techniques and previously designed parts. He/She fully understands CAD functions and can document system developments (parts lists, descriptions of function, operating instructions).	He/She can design and build autonomous mechatronic subsystems and, with suitable measuring and testing facilities, can assess the necessary production accuracy. He/She can document the results with quality-control systems.	He/She can make independent adaptations to the various devices (including selection of drives, sensors, SPS) and can use CNC programs for building the system. He/She can, through a digital mock up, assemble and simulate the functioning system and use computeraided computations (e.g. FEM). He/She can perform cost-benefi t analyses (e.g. as a basis for deciding whether components should be bought or individually constructed.)	He/She can independently develop complex mechatronic systems and can calculate the economic usefulness of the system. He/She can optimise CNC programs for the manufacturing of complex mechatronic devices and systems and monitor the automated quantity of an open loop control system.

5. Putting mechatronic systems into operation and providing clients with technical and economic support	He/She can, according to specifications and blueprints, put mechatronic devices into operation and provide support to the client in the handover phase.	He/She, after considering the enterprise's needs and basic conditions, can put the mechatronic systems into operation, create the necessary documentation, advise the customer on safe operations of the devices, and advise on future technology selection.	He/She, after considering all basic conditions, can master the start-up of interconnected mechatronic systems and machines, and can provide the necessary documentation including a manual. He/She can review client needs and confi gure machines that provide solutions. He/She can train the customer where necessary and provide support for safe operating procedures.	He/She can evaluate customer requirements for mechatronic facilities, develop solutions, and can plan the system's implementation and operation.	He/She can direct, including scheduling and time management, the start-up of the project from the creation of a proposal to the client's acceptance.
6. Supervising and evaluating both the process sequences of mechatronic systems and facilities and the operational sequence (including quality assurance	He/She can supervise process sequences according to specifi cations as well as implement any requested quality-control measures.	He/She can independently supervise the process sequences, evaluate the results, operate an accompanying statistic process control (SPC) for the quality control plan, and prepare simple work schedules, including production schedule and time management.	He/She can operate and supervise mechatronic facilities, choose testing and monitoring plans, set up the accompanying SPC, seek the optimal results of the production line according to material-flow, and provide work schedules including standard production times.	He/She can master the monitoring of complex mechatronic systems using virtual instruments and PPS systems as well as open loop control for the optimisation of machinery arrangement, material fl ow analysis, and scheduling.	He/She can optimise the process cycles of mechatronic production lines, provide instructions on modifying the PPS systems (e.g. adjustment to SAP systems) and introduce quality systems for continuous improvement processes (CIP/KVP).
7. Installing, configuring, programming and testing hardware and software components for control and regulation of mechatronic systems and facilities	He/She is able to install and configure programs for hardware and software components as well as set up simple software control programs (SPS).	He/She can master the selection of hardware and software for mechatronic systems (sensors, actuators, interfaces, communication procedures) and can provide and test simple software control programs (SPS) according to production process requirements.	He/She can integrate and confi gure program-, control-, and regulation-mechanisms in mechatronic systems, program simple devices (in co-operation with developers), and simulate the program sequence before start-up.	He/She can develop, test, and configure hardware and software solutions for networked mechatronic systems; and can monitor system conditions with suitable measuring and visualisation tools.	
8. Preparing and distributing the technical information for adjustment of each enterprise's mechatronic systems	He/She can provide descriptions and designs of mechatronic subsystems and is familiar with the basic CAD applications.	He/She can fully understand the management of technical information documents for mechatronic systems and can prepare and adapt these documents according to an enterprise's specific operating requirements.	He/She is able to analyse complex operational sequences separately in order to understand the connections and draw up maintenance and production procedures. He/She can understand that the system parameters are important for the equipments' functions and can independently assess and document the wear and general conditions of the mechatronic equipment.		
9. Diagnosing and repairing malfunctions with mechatronic systems and facilities, advising clients on avoiding malfunctions, and modifying and expanding mechatronic systems	He/She can diagnose and repair errors and malfunctions on the simple components and devices in the mechatronic systems. He/She can use the necessary checking, measuring, and diagnostic tools.	He/She can independently correct problems in mechatronic production equipment with the help of (computer-aided) diagnostic systems and the use of expert systems, databases, and error documentations.	He/She can diagnose and repair errors and disturbances in complex mechatronic equipment and is able to advise clients on how to avoid sources of malfunctions through changes or upgrades in the equipment and system.	He/She can diagnose and repair errors and disturbances in complex mechatronic equipment and is able to advise clients on how to avoid sources of malfunctions through changes or upgrades in the equipment and system.	

Berufsschule für Fertigungstechnik	**Contents / Learning Outcome**	PLC-Module

This table can be used

a) to locate the learning outcomes in the contents of the PLC-Module
b) and also the allocation within the Taxonomy Table

Example:

a) In Chapter 2.1 Hardware Config. Information the verbs understand and interpret are used to describe the learning outcomes.
b) The verbs indicate complexity 2 in the cognitive process dimensions, the types of knowledge are F which stands for factual knowledge and Ca which stands for causal knowledge.

	Contents		Learning Outcomes	Taxonomy Table
			1. Introduction PLC	
1.1	History of PLC	1.1.1	The S. is able to recognize important steplandmarks of the history of PLC	1F,
		1.1.2	S is able to recognize the reasons for the development of the PLC	1Ca
1.2	Difference of CPC-PLC	1.2.1	S is able to to recognize the difference between CPC-PLC	1F, 1Ca
			2. Modular PLC	
2.1	Hardware Config. Information	2.1.1	S is able to understand and interpret the meaning of hardware configuration	2F, 2Ca
2.2	Puzzle Modular PLC	2.2.1	S is able to carry out a standard hardware configuration by means of a puzzle	3P
2.3	Worksheet Modular PLC	2.3.1	S is able to recall the modules and their functions	1F, 1Ca
2.4	Hardware Configuration Station	2.4.1	S is able to make a list of modules mounted at their station (exemplify)	2F
2.5	Hardware Configuration	2.5.1	S is able to carry out HWK with SIMATIC Manager	3Ca
			3. Addressing	
3.1	Addressing Information	3.1.1	S is able to execute the addressing of DI and DO-modules	3Ca
		3.1.2	S is able to differentiate between DI and DO-modules and the necessary addresses,	4Ca
		3.1.3	S is able to check the addressing by means of the hardware configuration and the information given	5F

4. Programming				
4.1	Basic Bit Logic	4.1.1	S is able to recall the logic of OR, AND, SR and differentiate between them, using also the help function of the SIMATIC Manager	1F, 1Ca 4Ca
4.2	Program Exercise 1	4.2.1	S is able to understand why and how she/he use the logic functions	2P
		4.2.2	S is able to implement and organize simple programming	3Ca 4Ca, 4P
4.3	CPU Cycle Information	4.3.1	S is able to explain the CPU-cycle	2F, 2Ca, 2P
4.4	CPU Information	4.4.1	S is able to recall the modes of the CPU	1F, 1Ca
		4.4.2	S is able to choose and carry out the correct mode of the CPU	3Ca
4.5	Using the Glossary	4.5.1	S is able to understand how to use the Glossary of the SIMATIC Manager	2F
4.6	Using the Help Instruction	4.6.1	S is able to interpret the use of the Help Instruction.	2F, 2Ca, 2P
		4.6.2	S is able to carry out (work with) the information given in the help instruction.	3Ca, 3P
5. Analysing				
5.1	Variable Table	5.1.1	S is able to interpret the instruction for the use of the Variable Table	2F, 2Ca, 2P
		5.1.2	S is able to implement (use) the Variable Table to monitor and modify in- and outputs	3F, 3Ca, 3P
5.2	Symbol Table	5.2.1	S is able to recall the difference between the Variable and a Symbol Table	1P
		5.2.2	S is able to implement a Symbol Table in the existing programme	3P
5.3	Analyse Outputs	5.3.1	S is able to recall how to draw a pneumatic diagram	1F, 1Ca, 1P
		5.3.2	S is able to implement a pneumatic diagram	3Ca, 3P
		5.3.3	S is able to organise the movement of cylinders in a correct order manually, and with the Variable Table	4F, 4Ca, 4P
		5.3.4	S is able to complete and check the complete Symbol Table with the help of the Variable table	5F, 5Ca, 5P
5.4	Electrical circuit	5.4.1	S is able to understand (summarize) an electrical circuit	2F, 2Ca, 2P
		5.4.2	S is able to carry out the drawing of an electric circuit by means of an simulation program	3F, 3Ca, 3P
		5.4.3	S is able to differentiate between the functions of the parts of the electric circuit	4Ca, 4P

6. Sequence Chain				
6.1	Structured Program	6.1.1	S is able to recall the structure of a program	1F, 1Ca, 1P
		6.1.2	S is able to exemplify why a program should have a structure and how it can be structured	2Ca
6.2	Sequence chain	6.2.1	S is able to exemplify the principles of o a sequence chain	2P
		6.2.2	S is able to implement a sequence chain into FC 2	3Ca, 3P
		6.2.3	S is able to organize a program in different FCs	4P
		6.2.4	S is able to check and evaluate his own program	5F, 5Ca, 5P
6.3	Pushbuttons and switches	6.3.1	S is able to implement a standard set of pushbuttons and switches	3Ca, 3P
		6.3.2	S is able to organize the different functions of the mechatronic system by means of the switches and pushbuttons	4P
6.4	FC1 Modes of operation	6.4.1	S is able to compare the different modes of operation	2P
		6.4.2	S is able to carry out and organize the programming of the necessary networks for the modes of operation.	3P 4P
		6.4.3	S is able to check the correct operation of the modes of operation	5F, 5Ca, 5P
6.5	FC4 indication	6.5.1	S is able to carry out and organize the programming of the necessary networks for the indication lamps.	3P 4P
		6.5.2	S is able to check the programming of a network that Indicates an error	5F, 5Ca, 5P

Taxonomy Table for the PLC Module

		Cognitive Process					
		Remember (1)	Understand (2)	Apply (3)	Analyze (4)	Evaluate (5)	Create (6)
knowledge	Factual knowledge (F)	1.1.1 1.2.1 2.3.1 4.1.1 4.4.1 5.3.1 6.1.1	2.1.1 2.4.1 4.3.1 4.5.1 4.6.1 5.1.1 5.4.1	5.1.2 5.4.2	5.3.3	3.1.3 5.3.4 6.2.4 6.4.3 6.5.2	
	Casual knowledge (Ca)	1.1.2 1.2.1 2.3.1 4.1.1 4.4.1 5.3.1 6.1.1	2.1.1 4.3.1 4.6.1 5.1.1 5.4.1 6.1.2	2.5.1 3.1.1 4.2.2 4.4.2 4.6.2 5.1.2 5.3.2 5.4.2 6.2.2 6.3.1	3.1.2 4.1.1 4.2.2 5.3.3 5.4.3	5.3.4 6.2.4 6.4.3 6.5.2	
	Procedural knowledge (P)	5.2.1 5.3.1 6.1.1	4.2.1 4.3.1 4.6.1 5.1.1 5.4.1 6.2.1 6.4.1	2.2.1 4.6.2 5.1.2 5.2.2 5.3.2 5.4.2 6.2.2 6.3.1 6.4.2 6.5.1	4.2.2 5.3.3 5.4.3 6.2.3 6.3.2 6.4.2 6.5.1	5.3.4 6.2.4 6.4.3 6.5.2	

Contents of the PLC-Module
e. g. Chapter 5.3 Analyse Outputs can be found in different Cognitive processes and different types of knowledge.

Timetable for the Module

Average school day:

08.00 – 09.30	lessons	Room 02
09.30 - 09.45	morning break	
09.45 - 12.30	lessons	Room 02
13.30 - 15.30 or longer	Study, company visit, museum…	

school		
when	what	where
Su. 02.10.11	Students arrive in Munich	hostel
Mo. 03.10.11	09.00 meet and greet Organisation: tickets, meals, schedule,… City rallye	room 320
Tu. 04.10.11	08.00 lessons: Hr. Schauhuber, Hr. Schott 13.30 study	Room 02
We. 05.10.11	08.00 lessons: Hr. Schauhuber, Hr. Schott 13.30 company visit, SWM, Olympia swimming hall Hr. Lang, Hr. Stoll, Hr. Häfner	Room 02 SWM
Th. 06.10.11	08.00 lessons: Hr. Schauhuber, Hr. Schott 13.30 study	Room 02
Fr. 07.10.11	08.00 lessons: Hr. Schauhuber, Hr. Schott 13.30 Deutsches Museum, Hr. Kluger	Room 02 Dt. Museum
Sa. 08.10.11	Sightseeing Munich: Hr. Härder, Hr. Schwarz, Hr. Häfner, Meeting: ???	
Su. 09.10.11		
Mo. 10.10.11	08.00 lessons: Hr. Neumayr, Hr. Schott 13.30 study	Room 02
Tu. 11.10.11	08.00 lessons: Hr. Neumayr, Hr. Schott 13.30 company visit Seidenader: Hr. Rauchbart, Fr. Mittermaier, Hr. Häfner	Room 02 Seidenader
We. 12.10.11	08.00 lessons: Hr. Neumayr, Hr. Schott 13.30 study	Room 02
Th. 13.10.11	08.00 lessons: Hr. Neumayr, Hr. Schott, paper and pencil test: 10- 11h 12.00 lunch 12.53 departure Hackerbrücke 13.30 BMW museum, Hr. Bierbaum, Hr. Weber Hr. Bittner	Room 02 BMW
Fr. 14.10.11	08.00 lessons: Hr. Schauhuber, Hr. Schott, **test results** 9.00 Weißwurschtessen 11.00 be ready for international guests 14.00 sports: Frau Grundner, Hr. Schott	Room 02
Sa. 15.10.11		
Su. 16.10.11		

Average company day: example SWM

07.00	Meeting, than work: BMW, SWM	
08.00	Meeting, than work: Seidenader	
09.00-09.15	morning break	
12.00-12.45	Lunch break	
15.30	Leisure time	

Company: BMW, Seidenader, SWM **Example SWM**		
Mo. 17.10.11 Team work: mixed nation teams	Welcome, organisation, tour Programming in **teams**	BMW Seidenader SWM
Tu. 18.10.11 Team work:	Programming in **teams**	BMW Seidenader SWM
We. 19.10.11 morning afternoon single work	Programming in **teams** Additional Programming in **Single work**	BMW Seidenader SWM
Th. 20.10.11 Morning 8.00	Presentation, test Expert discussion of the work within the team, the company expert and the teacher	BMW Seidenader SWM
afternoon 13.00	Evaluation by Markus Müller TUM prepare for Friday: Presentation for Friday by spokespersons Fotos, reports, Katka, DVD	BSFT 02/05
Fr. 21.10.11 8.00 – app. 11.00 Celebration with partners	Certificates: given by companies Speeches, Presentations: students, BSFT, companies, TUM… Farewell and see you soon in …. Common lunch	BMW Riesen-feldstr. Tor 5

This action based training was developed within the Leonardo Da Vinci Transfer of Innovation Project:

"MODULES FOR VOCATIONAL EDUCATION AND TRAINING FOR COMPETENCES IN EUROPA"
"MOVET"
(PROJECTNUMBER **DE/10/LLP-LdV/TOI/147341)**

Module PLC

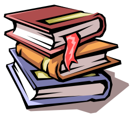
Lessons Material

The aim of the training is to enable the apprentices to develop the skills, knowledge and competence for competence area 7 of the competence Matrix Mechatronics from the VQTS model (cf. Karin Luomi-Messerer & Jörg Markowitsch, Vienna 2006)

7.3 He/She can integrate and configure program-, control-, and regulation-mechanisms in mechatronic systems, program simple devices (in co-operation with developers) and simulate the program sequence before start-up.

Berufsschule für Fertigungstechnik	1. Introduction PLC	PLC-Module

1.1 History



Origin

The PLC was invented in response to the needs of the American automotive manufacturing industry. Programmable controllers were initially adopted by the automotive industry where software revision replaced the re-wiring of hard-wired control panels when production models changed.
Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was accomplished using hundreds or thousands of relays, cam timers, and drum sequencers and dedicated closed-loop controllers. The process for updating such facilities for the yearly model change-over was very time consuming and expensive, as the relay systems needed to be rewired by skilled electricians.

In 1968 GM Hydramatic (the automatic transmission division of General Motors) issued a request for a proposal for an electronic replacement for hard-wired relay systems.
The winning proposal came from Bedford Associates of Bedford, Massachusetts. The first PLC, designated the 084 because it was Bedford Associates' eighty-fourth project, was the result. Bedford Associates started a new company dedicated to developing, manufacturing, selling, and servicing this new product: Modicon, which stood for MOdular DIgital CONtroller. One of the people who worked on that project was Dick Morley, who is considered to be the "father" of the PLC. The Modicon brand was sold in 1977 to Gould Electronics, and later acquired by German Company AEG and then by French Schneider Electric, the current owner.
One of the very first 084 models built is now on display at Modicon's headquarters in North Andover, Massachusetts. It was presented to Modicon by GM, when the unit was retired after nearly twenty years of uninterrupted service. Modicon used the 84 moniker at the end of its product range until the 984 made its appearance.
The automotive industry is still one of the largest users of PLCs.

Modicon 984

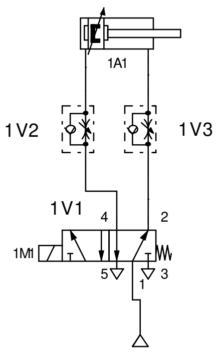
From Wikipedia, the free encyclopedia

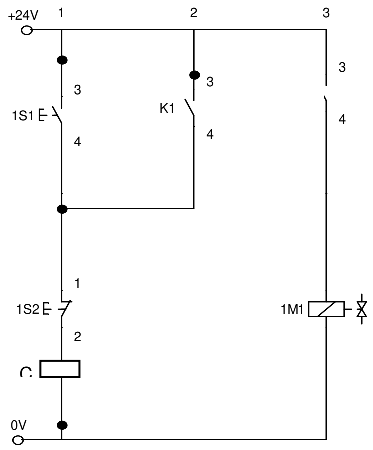
<u>Questions on the text 1.1:</u>
a) Why was the model change-over time-consuming and expensive?
b) What was the name of the first PLC and who is regarded as the father of it?

1.2 Difference CPC-PLC

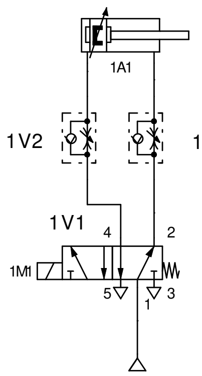
Before Programmable Logic Controllers were invented, the usual way to realize complex controls was relay technique.

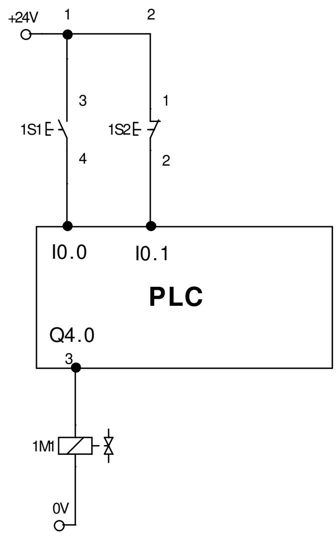
Connection programming control CPC
- Logic realized by connecting relays
- Changes consume time and money





Programmable Logic Control PLC
- The logic is realized with a programme in the PLC
- Changes can be made without changing the wiring

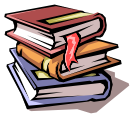




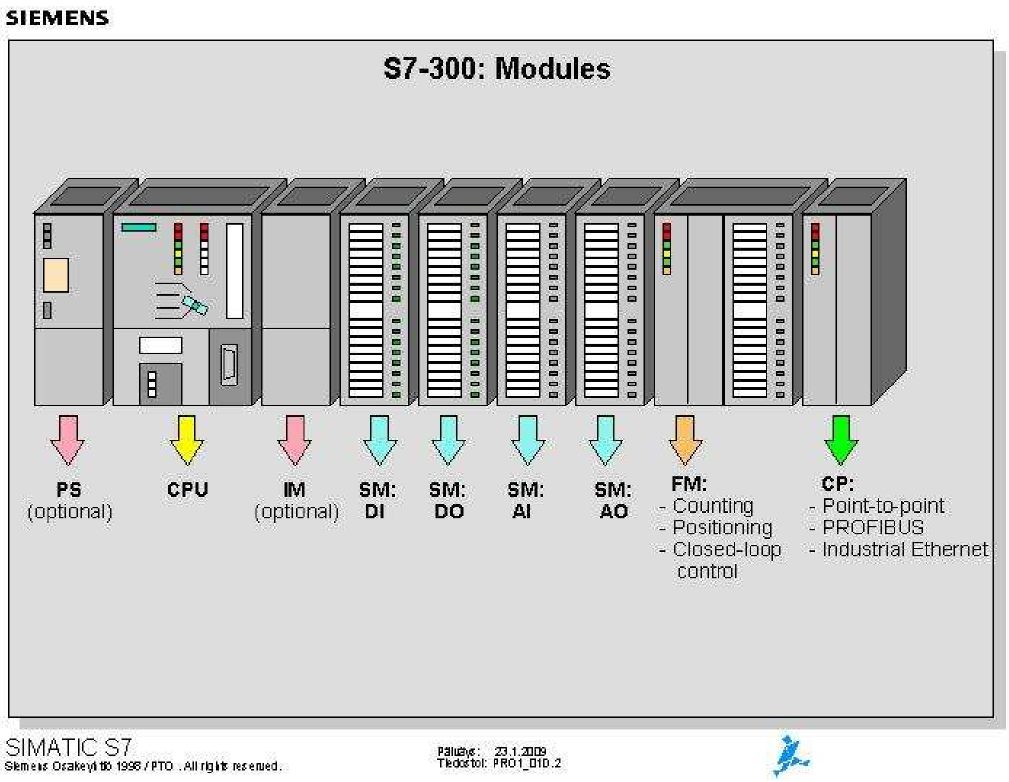
<u>Question on the text 1.2</u>
a) What is the difference between CPC and PLC?

Berufsschule für Fertigungstechnik	**2. Modular PLC**	PLC-Module

2.1 Information Hardware Configuration



Configuring
The term "configuring" refers to the arranging of racks, modules, distributed I/O (DP) racks, and interface submodules in a station window. Racks are represented by a configuration table that permits a specific number of modules to be inserted, just like a real rack.
In the configuration table (p. 5), STEP 7 automatically assigns an address to each module. You can change the addresses of the modules in a station if the CPU in the station can be addressed freely (meaning an address can be assigned freely to every channel of the module, independent of its slot).
You can copy your configuration as often as you like to other STEP 7 projects, modify it as necessary, and download it to one or more existing plants. When the programmable controller starts up, the CPU compares the preset configuration created in STEP 7 with the actual configuration of the plant. Any errors are therefore recognized immediately and reported.



Slot Numbers
The slot numbers in the rack of an S7-300 simplify addressing in the S7-300 environment. The position of the module in the rack determines the first address on a module.

Slot 1 Power supply. This is the first slot by default. A power supply module is not absolutely essential. An S7-300™ can also be supplied with 24V directly.

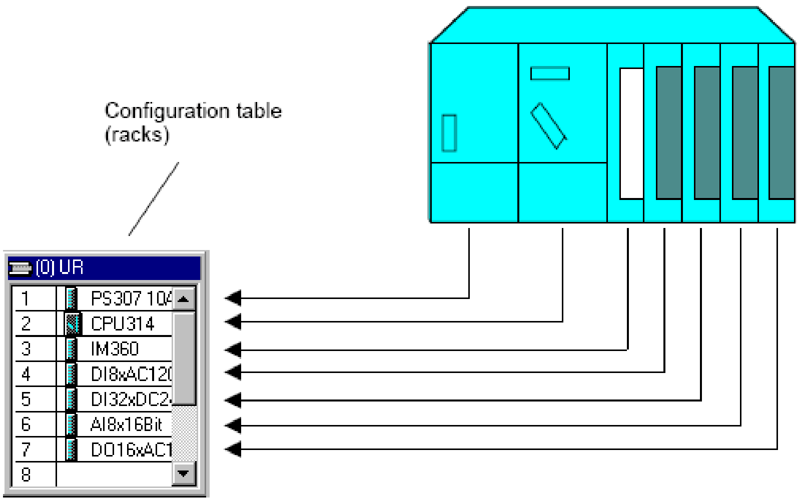
Slot 2 Slot for the CPU.

Slot 3 Logically reserved for an interface module (IM) for configurations using expansion racks. Even if no IM is installed, it must be included for addressing purposes. You can physically reserve the slot (such as for installing an IM at a later date) if you insert a DM370 dummy module.

Slots 4-11 Slot 4 is the first slot that can be used for I/O modules, communications processors (CP) or function modules (FM).
Addressing examples:
• A DI module in slot 4 begins with the byte address I0.0
• The top LED of a DO module in slot 6 is called Q8.0

Note Four byte addresses are reserved for each slot. When 16-channel DI/DO modules are used, two byte addresses are lost in every slot!

From: SIMATIC **Configuring Hardware and Communication Connections STEP 7 V5.2 Manual**
This manual is part of the documentation package with the order number: **6ES7810-4CA06-8BA0**
Edition 12/2002 A5E00171229-01



Questions on the text 2.1:

Explain the term "configuration"
What does a configuration table represent?
Which modules can be typically found in slot 1, 2, 3, 4, 5?
How many bytes are reserved for each slot?

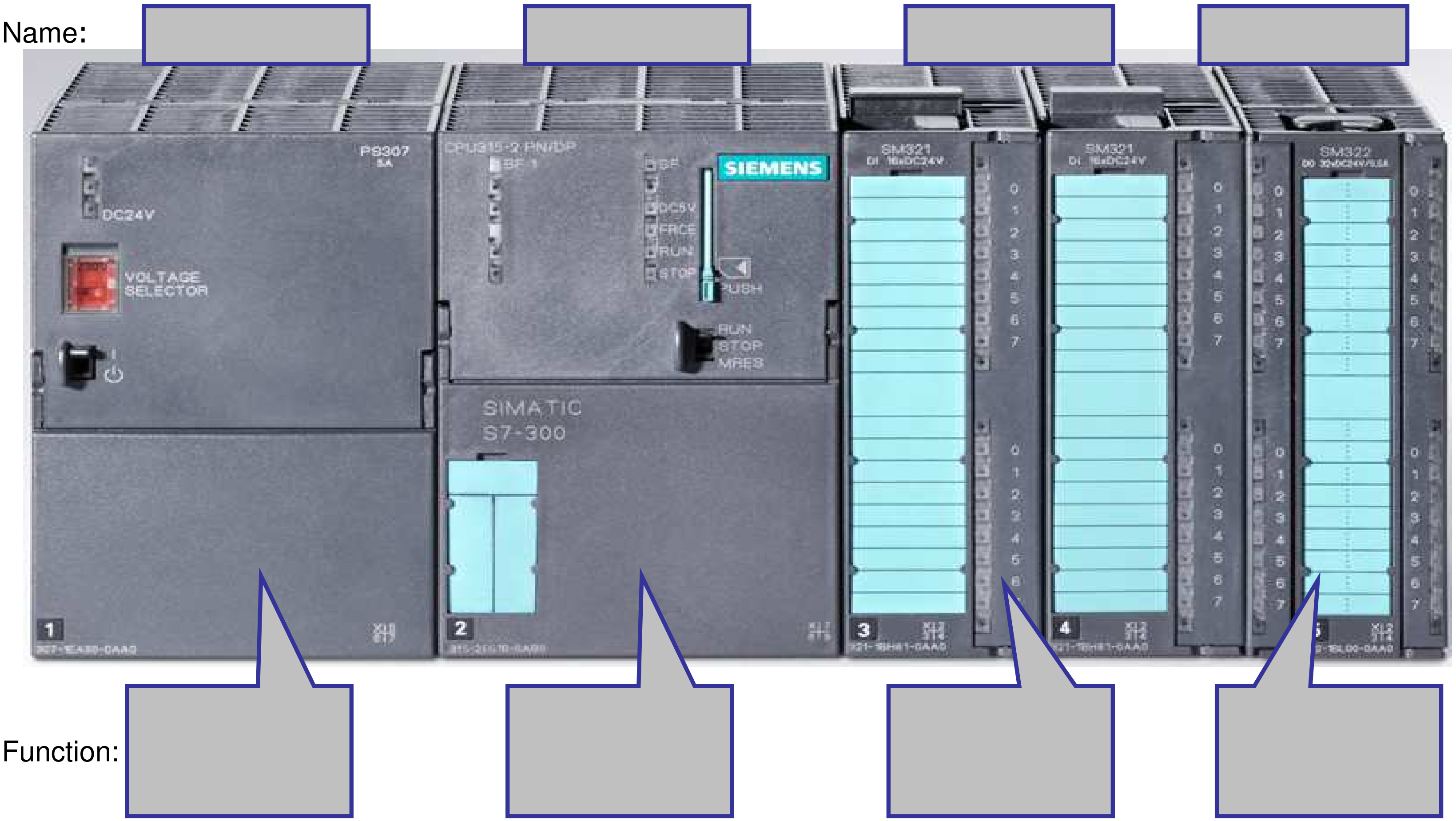
2.2 Puzzle

Solve the Puzzle Hardware Configuration with your partner.

2.3 Worksheet Modular PLC

Fill in the correct name and function for each module

Name:



Function:

2.4 Hardware Configuration Station

Work with your station.

Make a list of the S7 modules mounted at your station and describe their main functions.

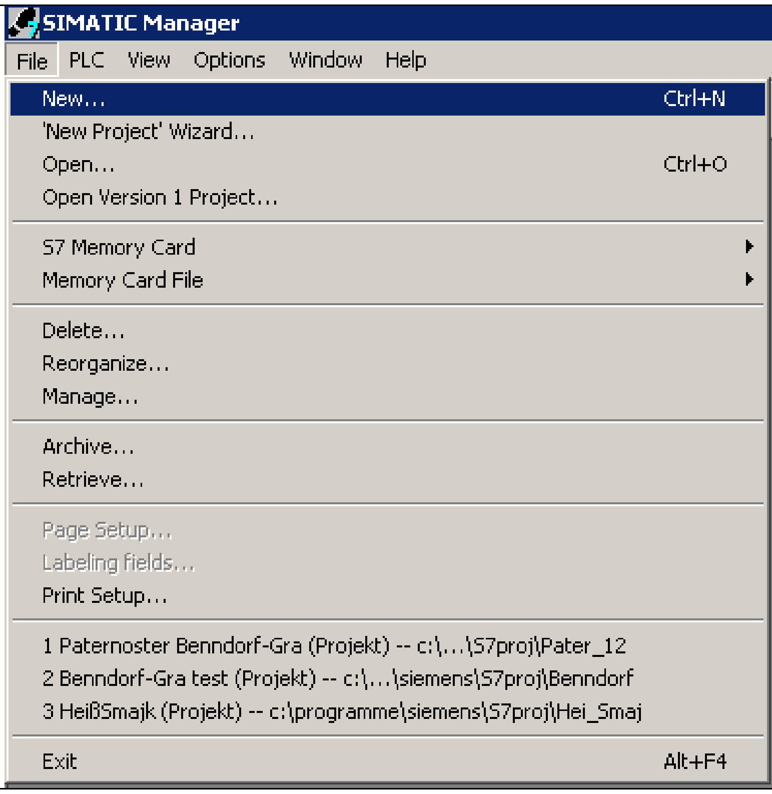
	Module name	Main Function
Example	*PS 307 Power Supply*	*Transfers 230V AC into 24V DC*
1		
2		
3		
4		
5		

Berufsschule für Fertigungstechnik	**2. Modular PLC**	PLC-Module

2.5 Hardware Configuration

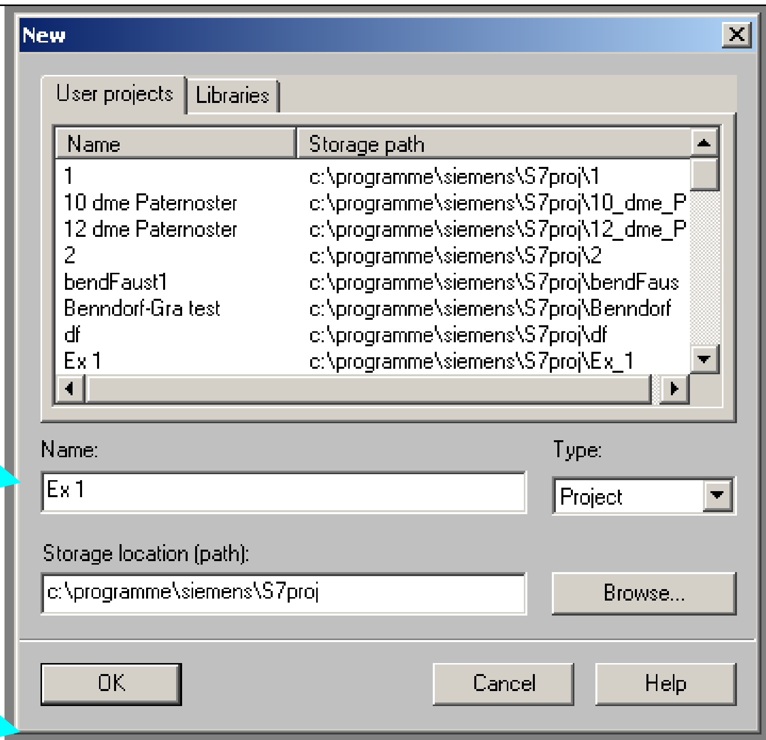
The Hardware Configuration is necessary to tell the programming device (e.g. PC) which modules it has to communicate with, similar to the procedure when a printer is installed into your PC.

To carry out the Hardware Configuration for your station, follow this procedure.

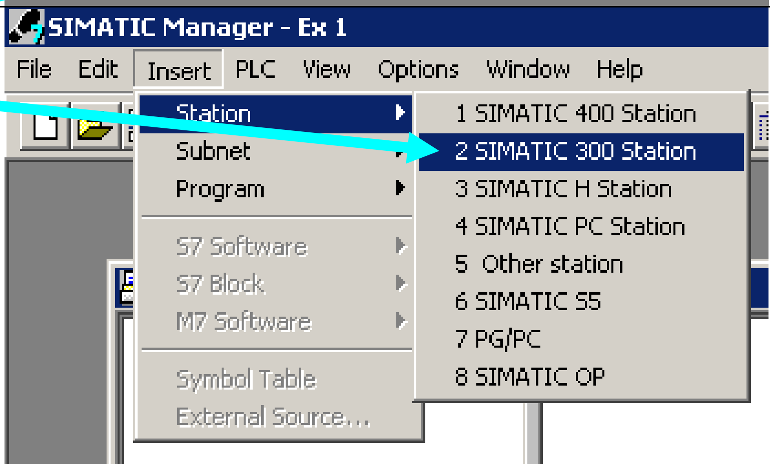
1. Start the SIMATIC-Manager	
2. Open a new file	

3. Name it “Ex 1”

4. Click Button “OK”



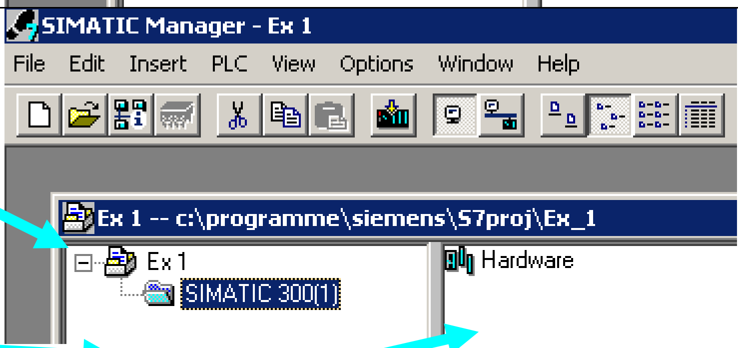
5. Insert a SIMATIC 300 Station



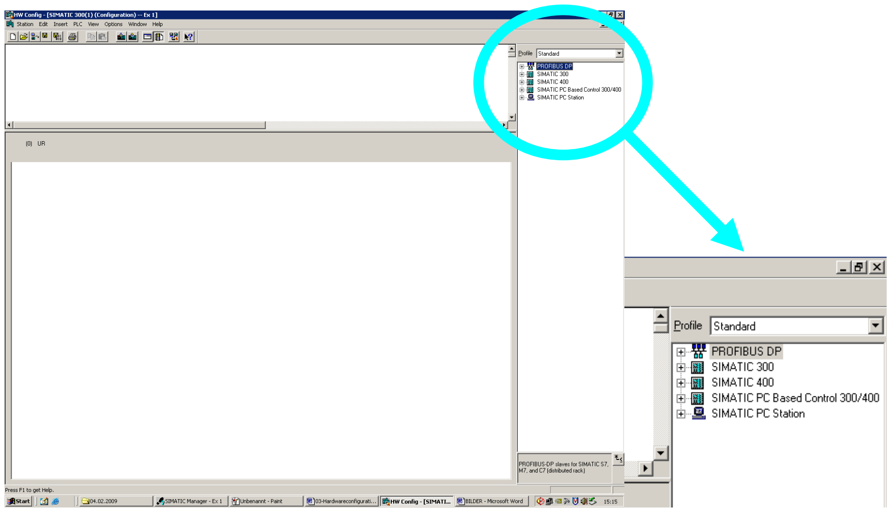
6. Open “⊟ Ex 1” folder

7. Click “SIMATIC 300(1)”

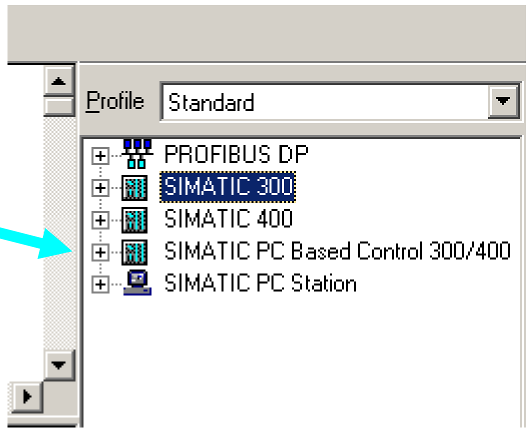
8. Double click on “Hardware”



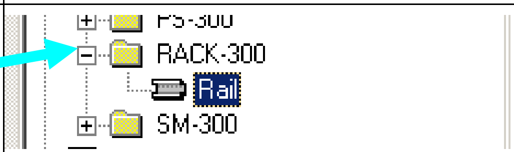
The window Hardware Configuration opens and under SIMATIC 300 all modules can be found.



9. Click "⊟ SIMATIC 300"

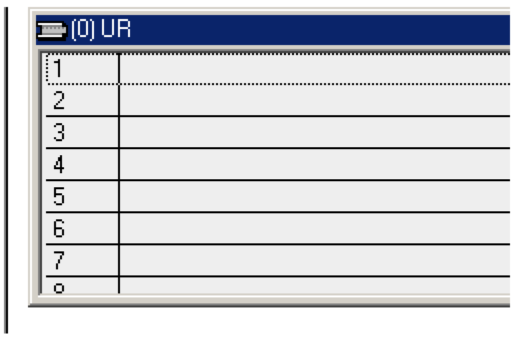


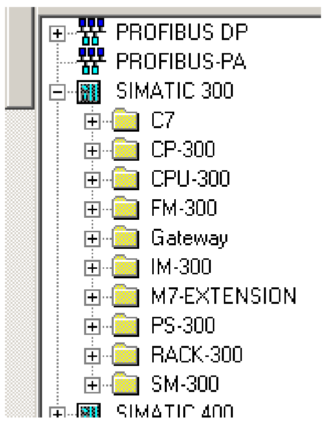
10. Double click on "⊟ Rail"



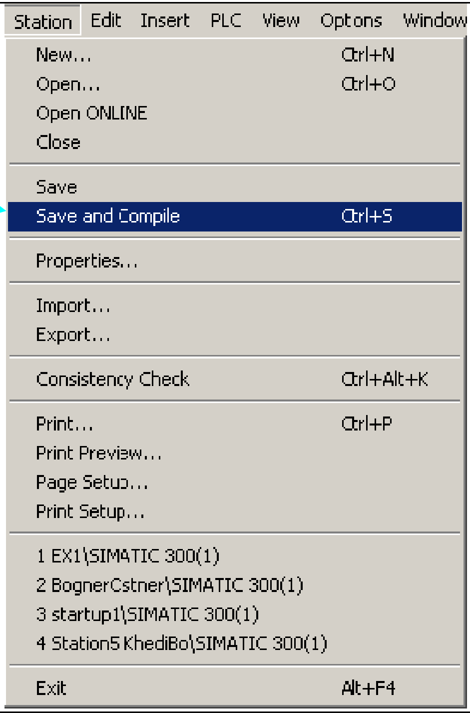
This window opens and if you mark one of the lines/slots a module can be inserted with a double click.

Usually all the necessary modules can be found under PS 300, CPU 300, SM 300.

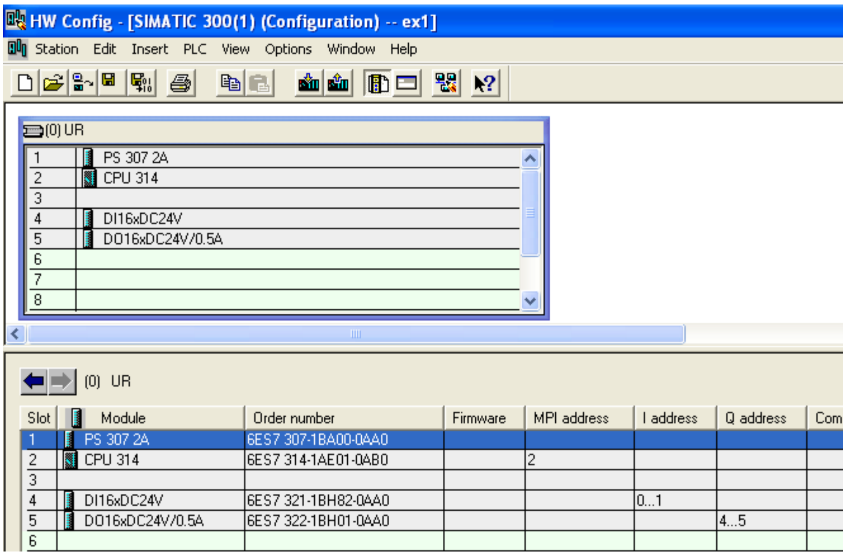




11. "Save and Compile" the station using the Station button and close the hardware configuration.



The result of your Hardware Configuration should be similar to this one:



<u>Question 2.5</u>
a) Which **Input** and **Output** addresses can be used with this project?
Please write them down:

____

____

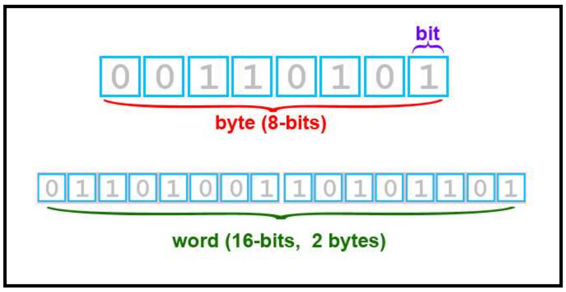
____

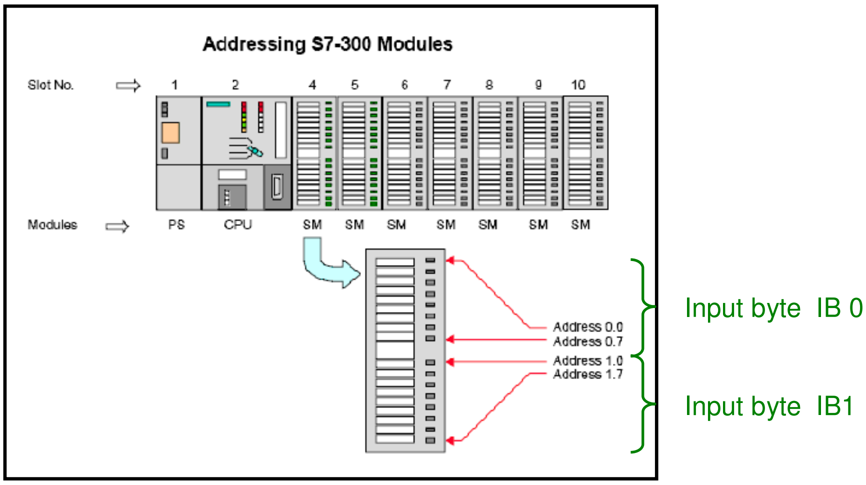
Berufsschule für Fertigungstechnik	**3. Addressing**	PLC-Module

3.1 Information Addressing

For every sensor or actuator, an in- or output address is necessary to connect it to your PLC. The addresses which can be used are organised in bits, bytes, words and so on.

bit – byte – word





Note: Four byte addresses are reserved for each slot. When 16-channel DI/DO modules are used, two byte addresses are lost in every slot!

Addressing examples:

- A DI module in slot 4 begins with the bit address I 0.0
- The top LED of a DO module in slot 6 is called Q 8.0

<u>Tasks 3.1:</u>

a) Fill in the correct addresses in 2.3 Worksheet Modular PLC (page 7)

b) What happens if you use a 16 bit DI module?

____

____

b) Your colleague programmes an output address “Q 1.9”.

____

4.1 Basic Bit Logic

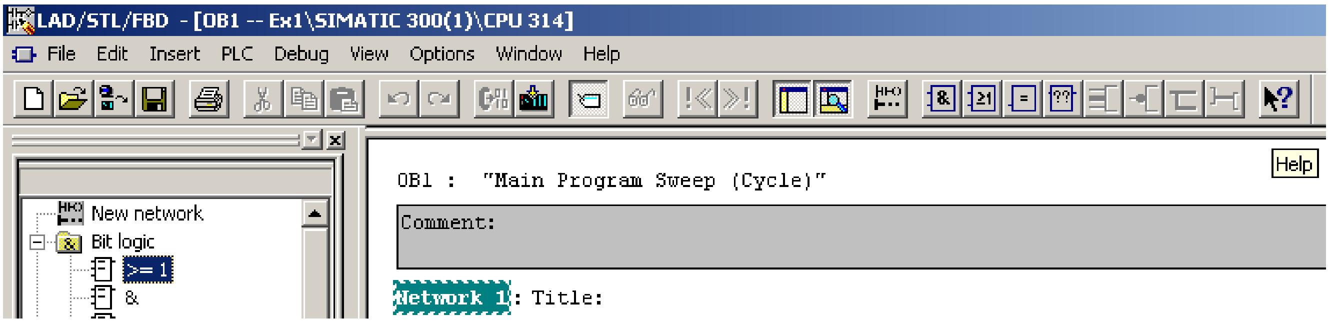
Function Block Diagram Programming Language (FBD)

The programming language Function Block Diagram (FBD) is based on graphic logic symbols also known in Boolean algebra. Complex functions such as maths functions can also be displayed directly in combination with the logic boxes.

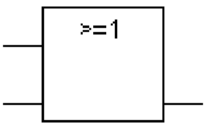
- Complete the function block symbols with in- and outputs.
- Complete the function table.
- Write a short description of the function.

Use the help function of the SIMATIC Manager to complete the function table:

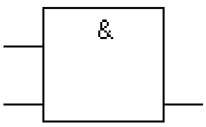
Click the ? button and click on the Bit Logic Symbol to activate the Help or use F1.



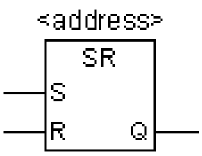
Basic Bit logic: OR, AND, SR

>=1 : OR Logic Operation	Function/truth table	Description of function
	see table below	

I1	I2	Q
0	0	
0	1	
1	0	
1	1	

& : AND Logic Operation	Function/truth table	Description of function
	see table below	

I1	I2	Q
0	0	
0	1	
1	0	
1	1	

SR : Set_Reset Flip Flop	Function/truth table	Description of function
	see table below	

I1	I2	Q
0	0	
0	1	
1	0	
1	1	

4.2 Programme Exercise 1

Open your project “Ex 1” with the existing hardware configuration and try to move the biggest actuator of your station.

For programming, use the logic functions: AND, OR and SR.

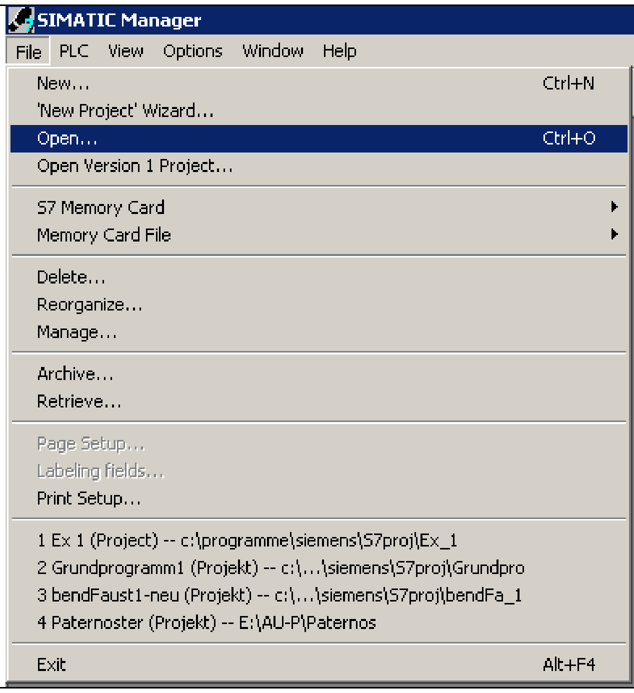
Choose two input addresses with contacts normally open and the output address of your biggest actuator (usually cylinder). Insert them in the yellow box in the Symbol table below like the Example in line 1.

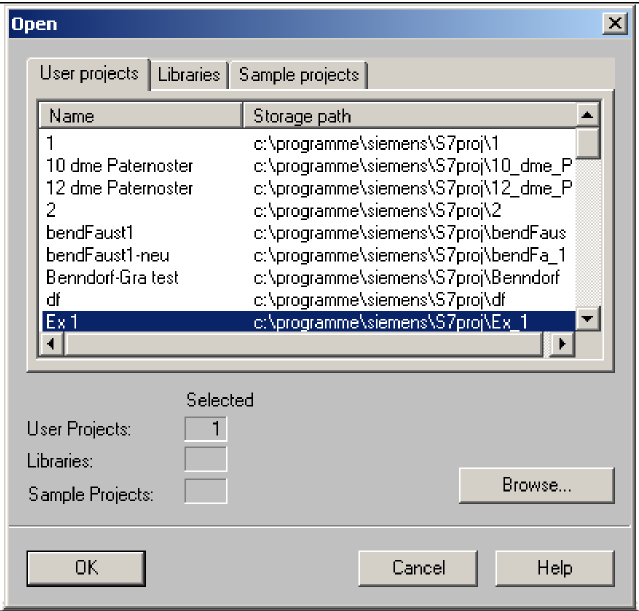
Symbol Table

	Symbol	Address	Data type	Comment
1	K1	Q 4.0	BOOL	conveyor start
2				biggest actuator
3				contact n.o.1
4				contact n.o.2
5				
6				
7				
8				

Open your project in the Simatic Manager where you have already completed the Hardware Configuration for your station and follow the procedure.

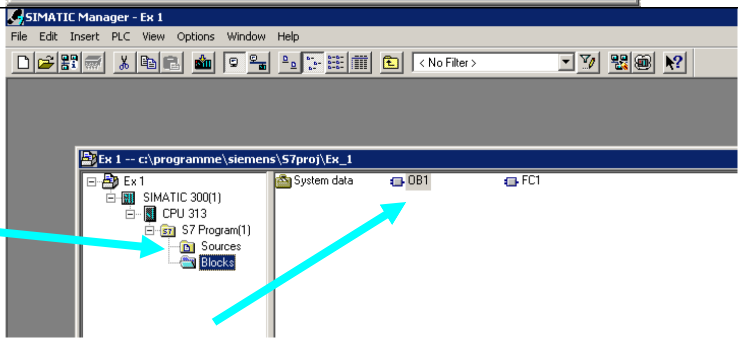
1. Open the project “Ex 1“





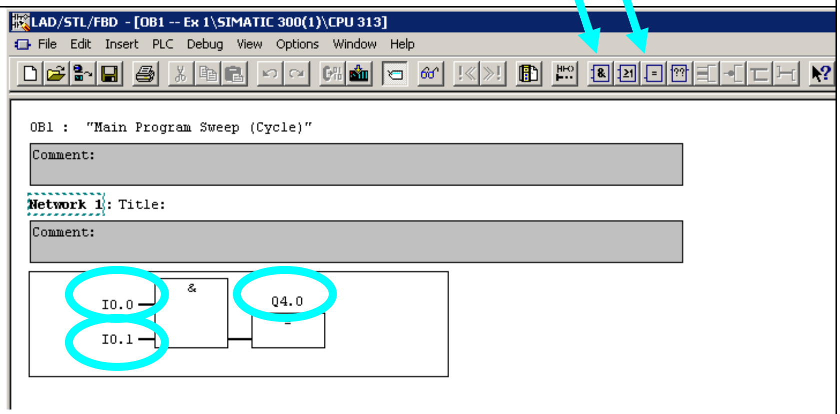
2. Press Button OK

3.Open “⊟ Ex 1” folder and mark “Blocks”



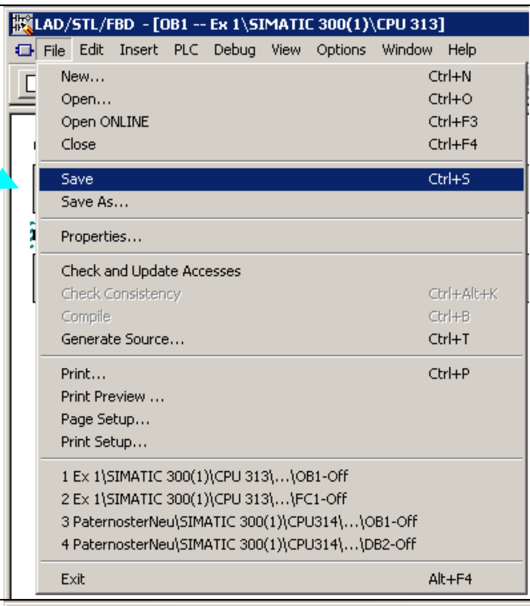
4. Double click on “OB1”

5. Click on & and =

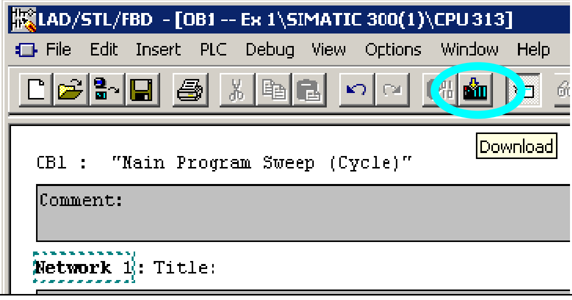


6. Name the addresses of input and output.
Make sure that you have used the correct addresses for your station that you have chosen in the Symbol Table above.

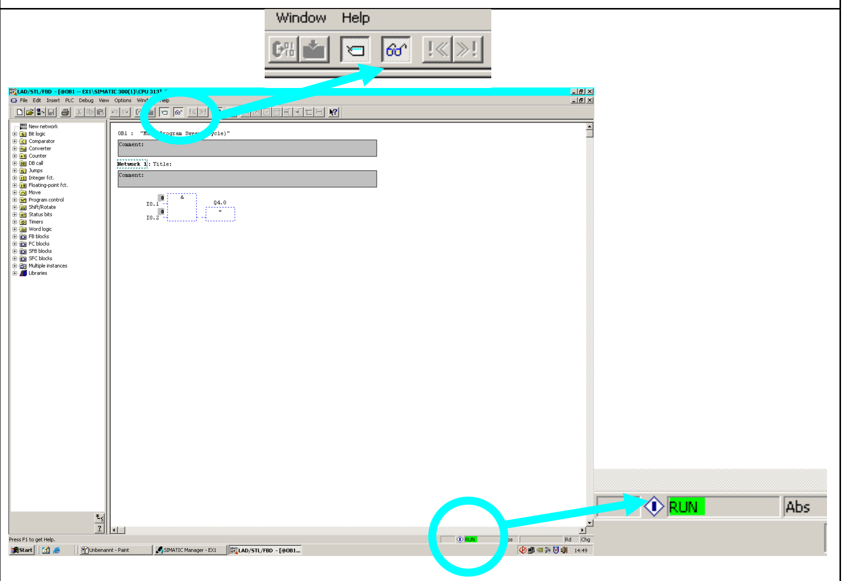
7. Save the project



8. Download and test your program



9. Monitor it while it is running



Execute the procedure with “OR” and “SR” in the same way.

Questions on 4.2

a) What happens if you have a signal at S and R at the same time?

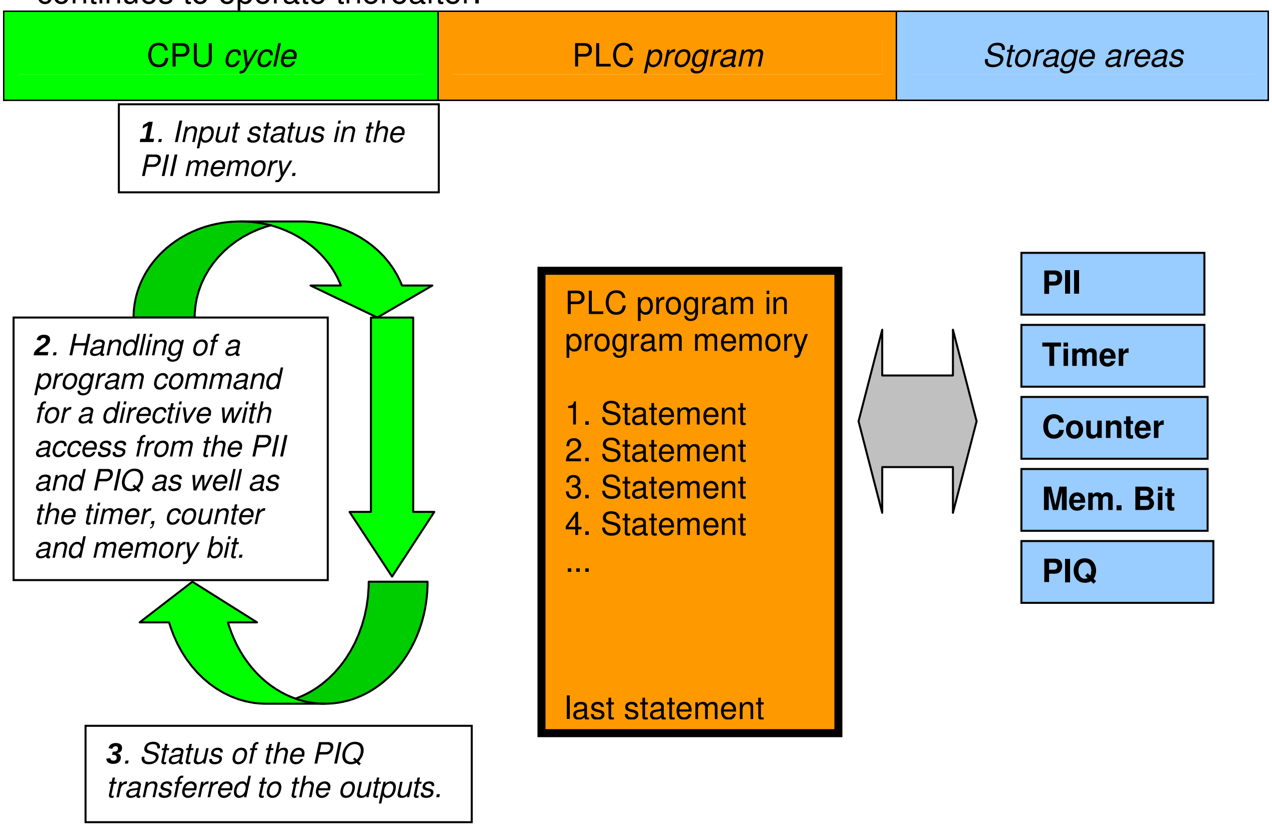
b) What happens if you have a short signal at S?

4.3 Information CPU Cycle

HOW DOES THE PROGRAM WORK IN A PLC?

Program processing in a PLC occurs cyclically with the following execution:

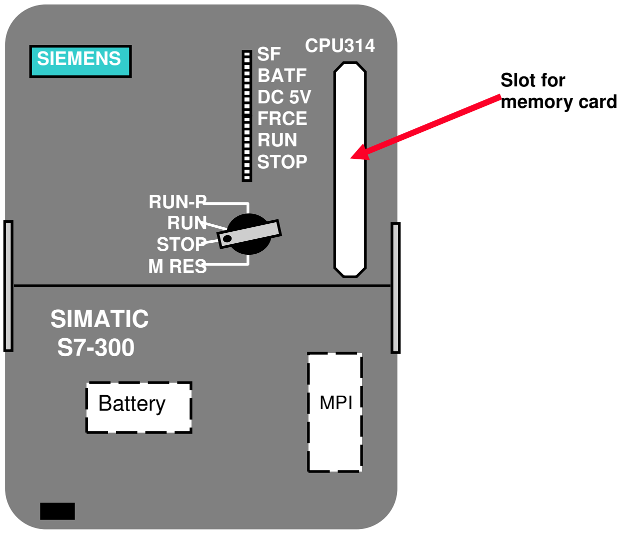
1. **Reading PII**: After the PLC is switched on, the processor (which represents the brain of the PLC) questions if the individual inputs have a 1 signal or not. This status of the input is stored in the process- image input table (PII). Leading inputs receive the information 1 or high when enabled, or the information 0 or low when not enabled.
2. **Processing the program:** This processor processes the program deposited into the program memory. This consists of a list of logic functions and instructions, which are successively processed, so that the required input information will already be accessed before the read in PII and the matching results are written into a process- image output table (PIQ). Also other storage areas for counters, timers and memory bits will be accessed during program processing by the processor if necessary.
3. **Transfer PIQ:** In the third step after the processing of the user program, the status from the PIQ will transfer to the outputs and then be switched on and/or off, it continues to operate thereafter.



<u>Questions on the text 4.3:</u>

a) What can be found in a PII?
b) What can be found in a PIQ?
c) Describe the three steps of a CPU cycle?
d) Where is the program that you have written and how is it processed?
e) Which other storage areas will be accessed during the CPU cycle?

4.4 Information CPU



Mode Selector (switch)
MRES = Memory reset function (**M**odule **Res**et)
STOP = Stop mode, the program has not been executed. A program can be transferred from the PG.
RUN = Program execution, read-only access possible from PG. No program transfer from PG.
RUN-P = Program execution, read/write access possible from PG.

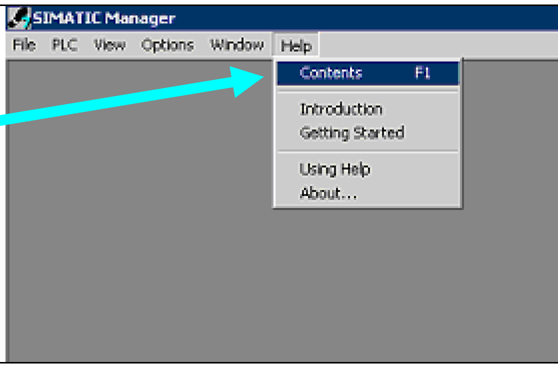
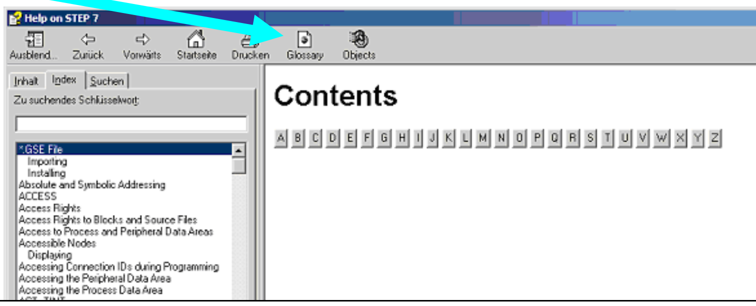
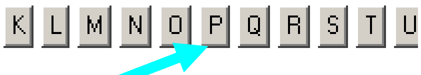
Status Indicators (LEDs)
SF = Group error; internal CPU fault or fault in module with diagnostics (LEDs) capability.
BATF = Battery fault; battery empty or missing.
DC5V = Internal 5 VDC voltage indicator.
FRCE = FORCE; indicates that at least one input or output is forced.
RUN = Flashes when the CPU is starting up, then a steady light in Run mode.
STOP = Shows a steady light in Stop mode.
Flashes slowly for a memory reset request,
Flashes quickly when a memory reset is being carried out,
Flashes slowly when a memory reset is necessary because a memory card has been inserted.

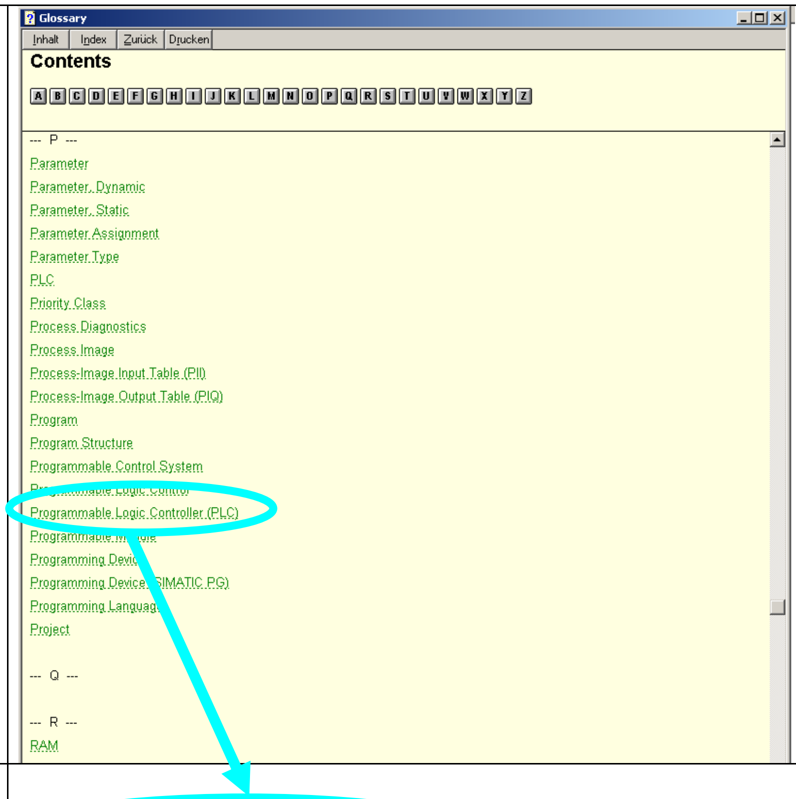
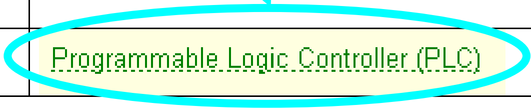
Memory Card
A slot is provided for a memory card. The memory card saves the program contents in the event of a power loss without the need for a battery.
For CPUs after Oct. 2002, a Micro Memory Card is always necessary for operation. It also provides the backup in the event of a power loss.

MPI- Interface:
Each CPU possesses an MPI interface for the networking of program devices (e.g. PC adapter). This is found behind a flap at the front of the CPU.

<u>Questions on the text 4.4:</u>
a) Which modes are possible if you want to execute a program?
b) Which mode do you have to select if you want to have writing access from the PG and execute a program?
c) Describe the status of the LEDs when a program is transfered from the PG to the CPU before it becomes fully active.
d) What is connected to the MPI-interface?
e) Describe the situation where a Memory Card is very useful.

4.5 Using the Glossary

<table>
<tr><td>1. Start the SIMATIC-Manager</td><td colspan="2">
</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>2. Open Contents or press F1</td><td colspan="2">
</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>3. Choose the Glossary</td><td colspan="2">
</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>4. Press P</td><td>
</td><td>Example for “PLC”</td></tr>
</table>

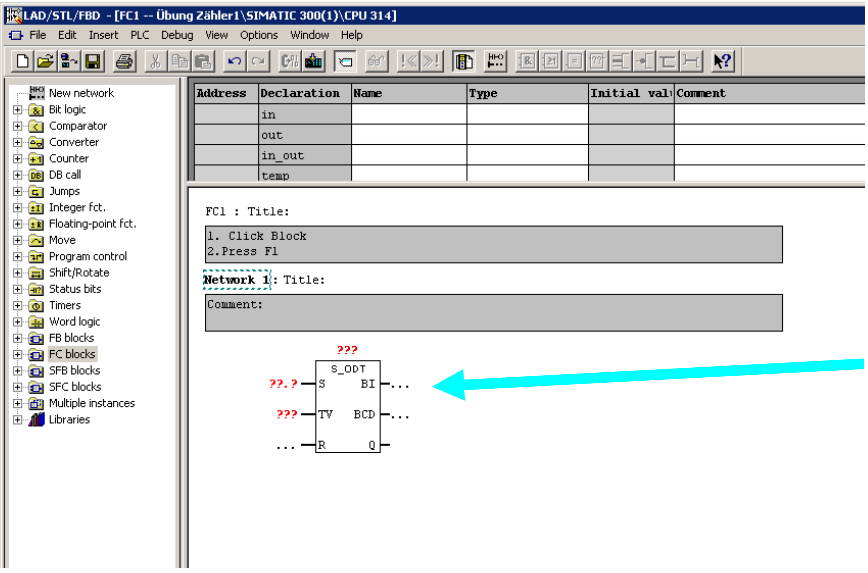
5. Choose the correct phrase.	
	

<u>Task 4.5</u>
Find out the following abbrevations:

PLC	
PII	
PIQ	
CPU	

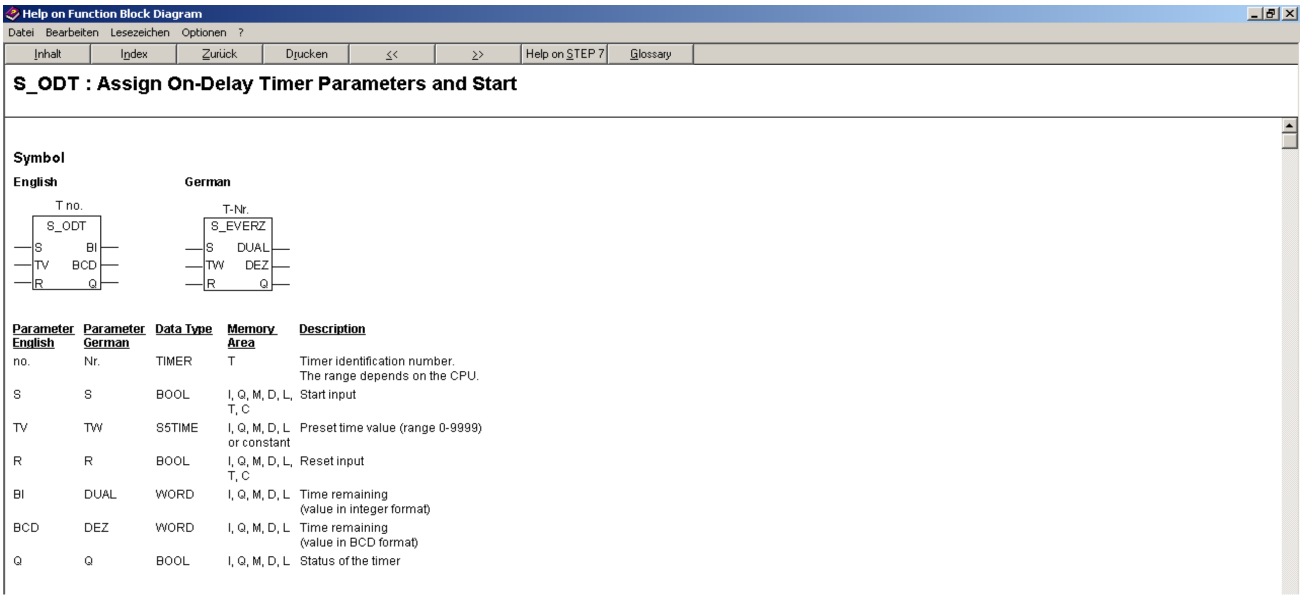
Berufsschule für Fertigungstechnik	4. Programming	PLC-Module

4.6 Using Help Instruction



1. Mark unknown symbol

2. Press **F1** to get information about the unknown symbol


Help on Function Block Diagram

Datei Bearbeiten Lesezeichen Optionen ?

Inhalt | Index | Zurück | Drucken | << | >> | Help on STEP 7 | Glossary

S_ODT : Assign On-Delay Timer Parameters and Start

Symbol

English (T no., S_ODT: S, TV, R, BI, BCD, Q) — **German** (T-Nr., S_EVERZ: S, TW, R, DUAL, DEZ, Q)

Parameter English	Parameter German	Data Type	Memory Area	Description
no.	Nr.	TIMER	T	Timer identification number. The range depends on the CPU.
S	S	BOOL	I, Q, M, D, L, T, C	Start input
TV	TW	S5TIME	I, Q, M, D, L or constant	Preset time value (range 0-9999)
R	R	BOOL	I, Q, M, D, L, T, C	Reset input
BI	DUAL	WORD	I, Q, M, D, L	Time remaining (value in integer format)
BCD	DEZ	WORD	I, Q, M, D, L	Time remaining (value in BCD format)
Q	Q	BOOL	I, Q, M, D, L	Status of the timer

It is usually very helpful to read the description and study the example and the timing diagram.
Note: The timing diagram has two signal states 0 or 1.

Description

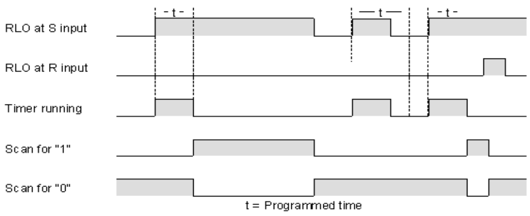
The **Assign On–Delay Timer Parameters and Start** instruction starts a specified timer if there is a rising edge (change in signal state from 0 to 1) at the Start (S) input. A signal change is always necessary to start a timer. The timer continues to run for the time specified at the Time Value (TV) input as long as the signal state at input S is 1. A signal state check for 1 at output Q produces a result of 1 when the time has elapsed without error and when the signal state at input S is still 1. When the signal state at input S changes from 1 to 0 while the timer is running, the timer is stopped. In this case, a signal state check for 1 at output Q always produces the result 0.

A change from 0 to 1 at the Reset (R) input of the timer while the timer is running resets the timer. This change also resets the time and the time base to zero. The timer is also reset if the signal state is 1 at the R input while the timer is not running.

The current time value can be scanned at the outputs BI and BCD. The time value at BI is in binary format; at BCD it is in binary coded decimal format.

Timing Diagram

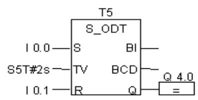
On-Delay timer characteristics:



Status Word

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
writes	-	-	-	-	-	X	X	X	1

Example



If the signal state of input I0.0 changes from 0 to 1 (rising edge in the RLO), timer T5 is started. If the specified time of two seconds (2s) elapses and the signal state of input I0.0 is still 1, the signal state of output Q4.0 is 1. If the signal state of input I0.0 changes from 1 to 0, the timer is stopped and output Q4.0 is 0. If the signal state of input I0.1 changes from 0 to 1 while the timer is running, the timer is restarted.

Start | Diplomarbeit | SIMATIC Manager - Ex 1 | LAD/STL/FBD - [FC1 -- Ü... | Dokument1 - Microsoft W... | **Help on Function Block...** | 16:44

<u>Task 4.6</u>
Study the example **Assign On-Delay Timer S_ODT** using the Help function and answer the following questions:
a) What is a rising edge?
b) What is falling edge?

Set your specified time of two seconds at TV (time value)
What happens at the output Q?
How long do you have a signal at Q ...
c) … if the signal at S is 8 seconds long?
d) … if the signal at S is shorter than 2 seconds?
e) … if there is a signal at input R after 3 seconds?

Berufsschule für Fertigungstechnik	**5. Analysing**	PLC-Module

5.1 Variable Table

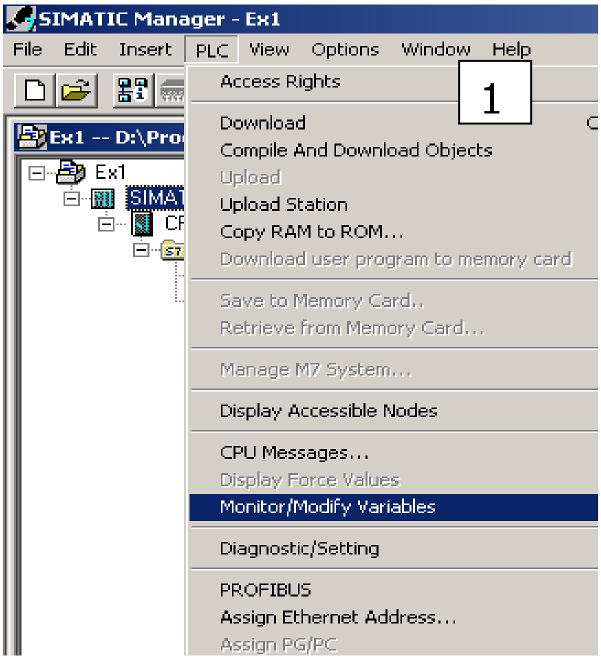
The Variable Table is helpful to check the electric wiring and to determine the addresses of the in- and outputs (variables). It has two main functions:

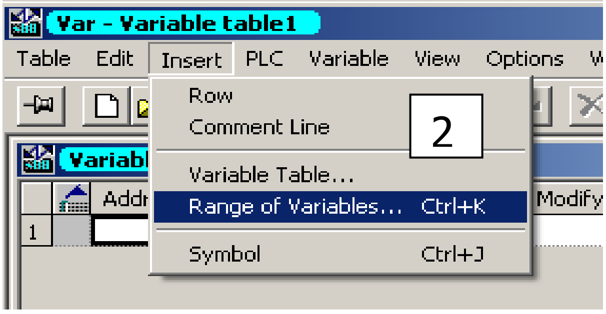
- With the Variable Table, the **inputs** (e.g. sensors and switches) can be **monitored**.
- With the Variable Table, the **outputs** can be **modified** (set).

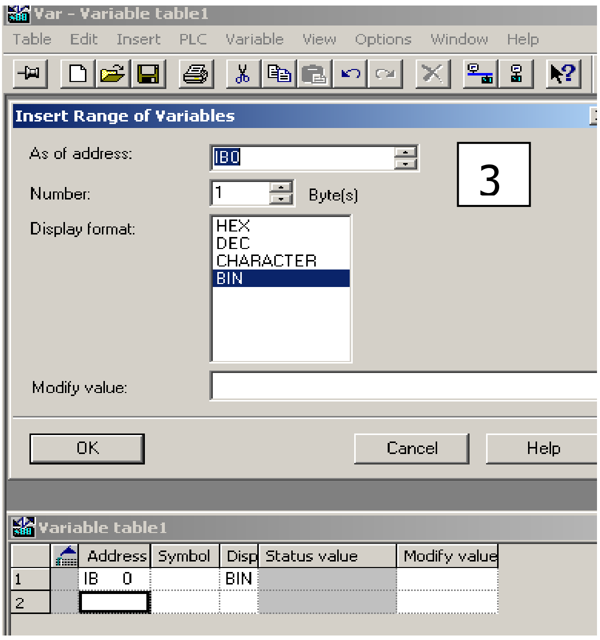
Monitor Inputs:

1. Open your project Ex1, choose PLC and Monitor/Modify Variables.
2. Insert a Range of Variables.
3. If you insert IB0 and choose BIN (binary) the whole input byte 0 can be monitored.
4. Monitor it with the help of the glasses button.

Note: The project with the Variable Table has to be downloaded to the CPU.







a)

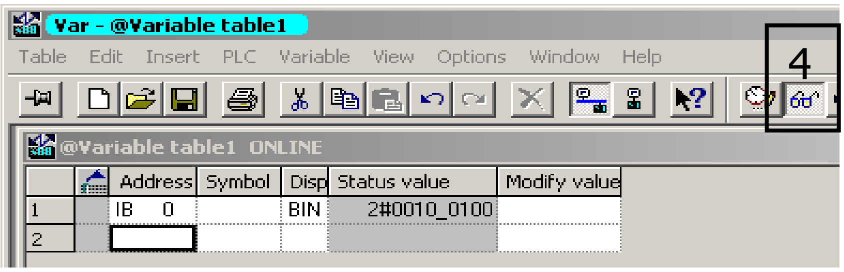
Which input bits from IB 0 have a 1-signal?

____

Your next task is to find out the **input addresses** of all switches, sensors etc. which are connected to the DI-Module of your station.

Insert all the input bytes of your station into the Variable Table (see above).

Activate or deactivate all sensors and switches (manually) and watch the status value.



Document input addresses in the **Symbol Table**
After you have checked all input addresses that are used in your station, write them down in the **Symbol Table** including comment.
Note: We use **S** for switches, and **B** for sensors.

Symbol Editor - [S7 Program(1) (Symbols) -- Ex1-1\SIMATIC 300(1)\CPU 314]

Symbol Table Edit Insert View Options Window Help

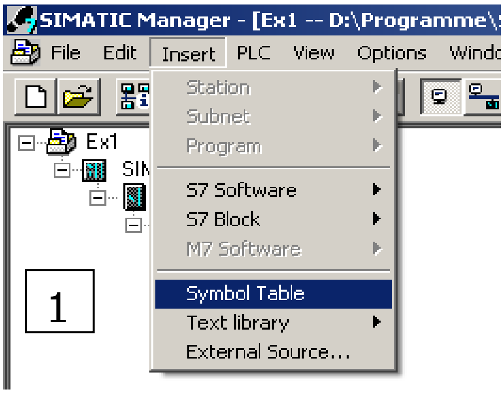
	Symbol	Address	Data type	Comment
1	1S0	I 0.0	BOOL	Start, push button normally open
2	1B1	I 0.1	BOOL	Cylinder 1A1, inital position
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15	K1	Q 4.0	BOOL	Motor, conveyor belt
16	1M1	Q 4.1	BOOL	Cylinder 1A1 extends
17				
18				
19				
20				
21				
22				
23				
24				
25				

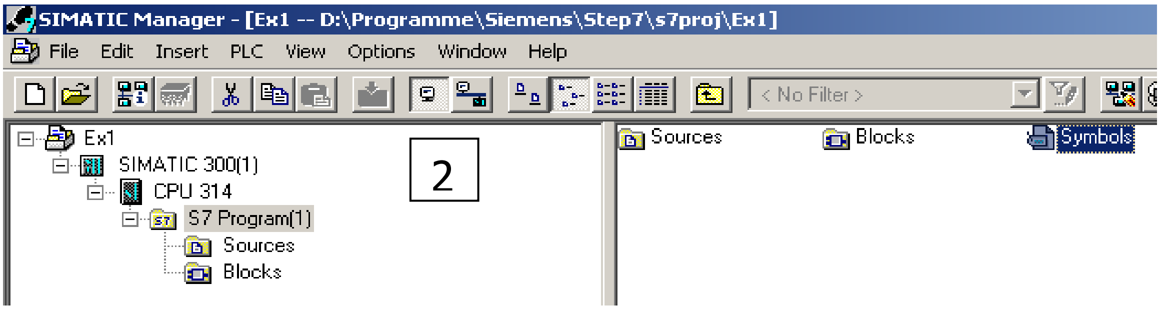
Task 5.1
b) What are the two main functions of the Variable Table?

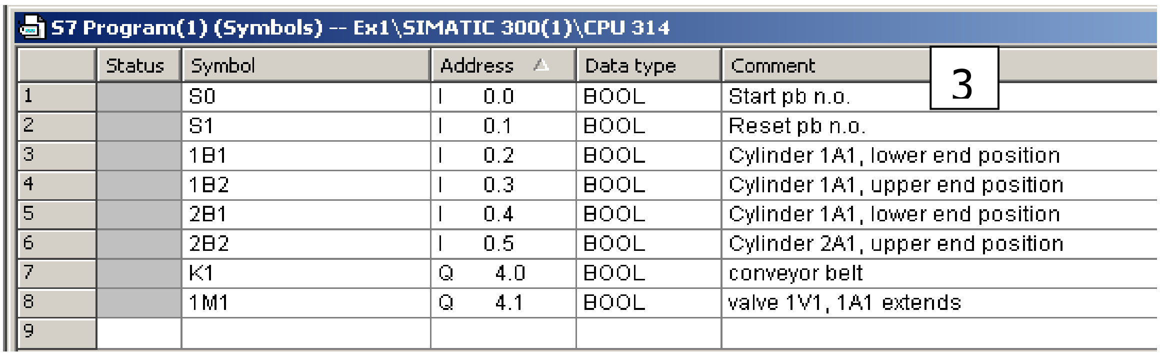
5.2 Symbol Table

The Symbol Table is a document for all the addresses that you use in your project. Usually, every address that we use has a symbolic name in the electrical circuit.
It is often easier to use symbols for programming instead of addresses.

Insert the Symbol Table with the SIMATIC Manager.
The Symbol Table is in the S7 Program level. Open Symbols.
Fill in all your in- and output addresses with the symbol and a comment.






S7 Program(1) (Symbols) -- Ex1\SIMATIC 300(1)\CPU 314

	Status	Symbol	Address		Data type	Comment
1		S0	I	0.0	BOOL	Start pb n.o.
2		S1	I	0.1	BOOL	Reset pb n.o.
3		1B1	I	0.2	BOOL	Cylinder 1A1, lower end position
4		1B2	I	0.3	BOOL	Cylinder 1A1, upper end position
5		2B1	I	0.4	BOOL	Cylinder 1A1, lower end position
6		2B2	I	0.5	BOOL	Cylinder 2A1, upper end position
7		K1	Q	4.0	BOOL	conveyor belt
8		1M1	Q	4.1	BOOL	valve 1V1, 1A1 extends
9						

Your Symbol Table is an important document when programming, make sure that there are no mistakes in it.

Question 5.2:
a) What is the difference between the Variable Table and the Symbol Table?

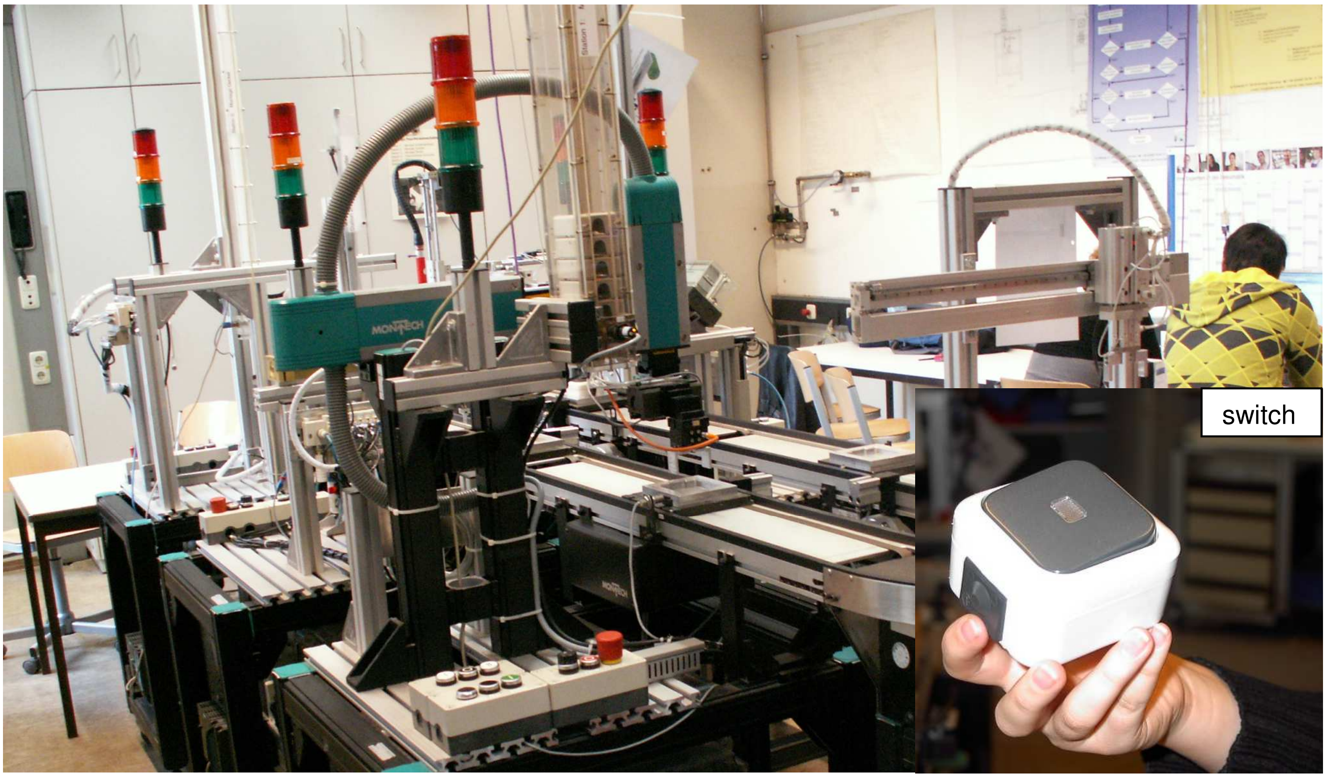
____

____

Berufsschule für Fertigungstechnik	**5. Analysing**	PLC-Module

5.3 Analyse Outputs

The mechatronic system you are working on, consists of 6 different stations.
It assembles a switch that consists of 5 different parts. Each station has a certain task and needs to be programmed separately by a team of students.



Task 5.3

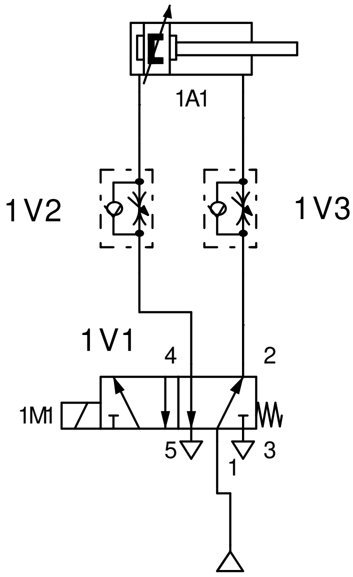
a) Analyse the correct movement of the cylinders for your task, and write it down step by step in the table below.

Note: Every movement in or out is 1 step.

Step	*Actuator / action*		
Example 1	*Cylinder 1A1 moves out*		
1			
2			
3			
4			
5			
6			
7			

Pneumatic Diagram

b) Draw a pneumatic circuit diagram of your station with the program FluidSIM. Try to be as accurate as possible.



Example with 1 actuator

MOVE IT !

c) Simulate the movements of your station manually.

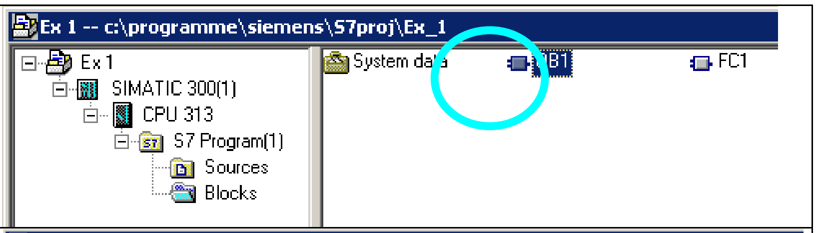
Every pneumatic valve at your station can be operated manually.

Label the valves and the actuators of your station according to your pneumatic diagram
(1V1, 1A1…) 1A1 1V1

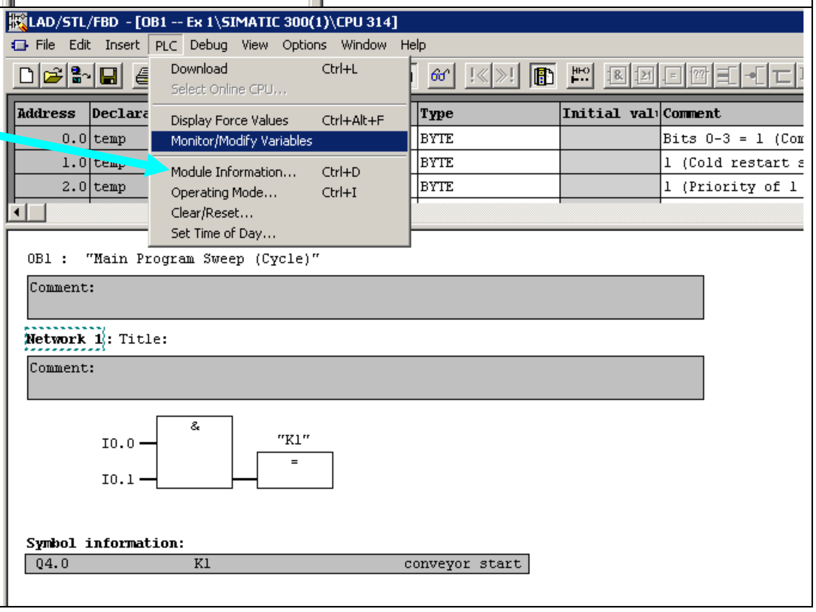
Now that you have checked all the movements your station performs, your next task is to find out how your valves/relays are connected to your DO Module at your PLC.

d) Variable Table
The Variable Table can be used to test this. Create a Variable Table for your outputs with the SIMATIC Manager:

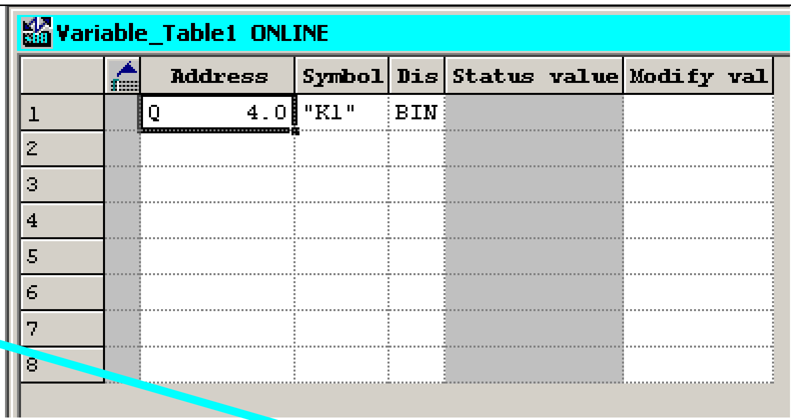
Open “OB 1”



Open “PLC” and click on “Monitor/Modify Variables”

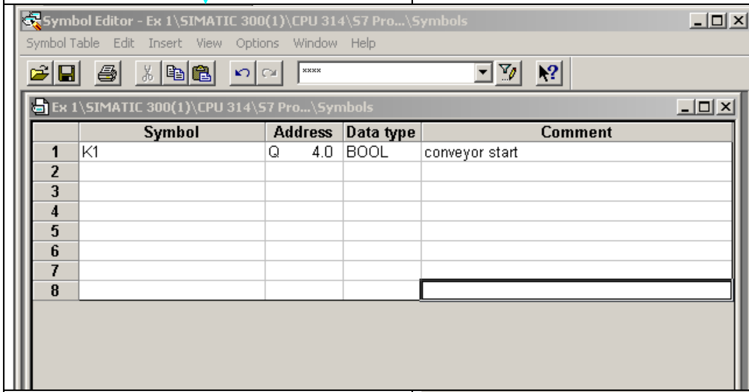


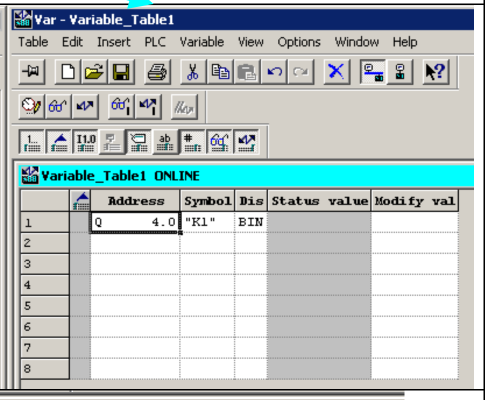
List your output addresses



Place the variable table next to your symbol table

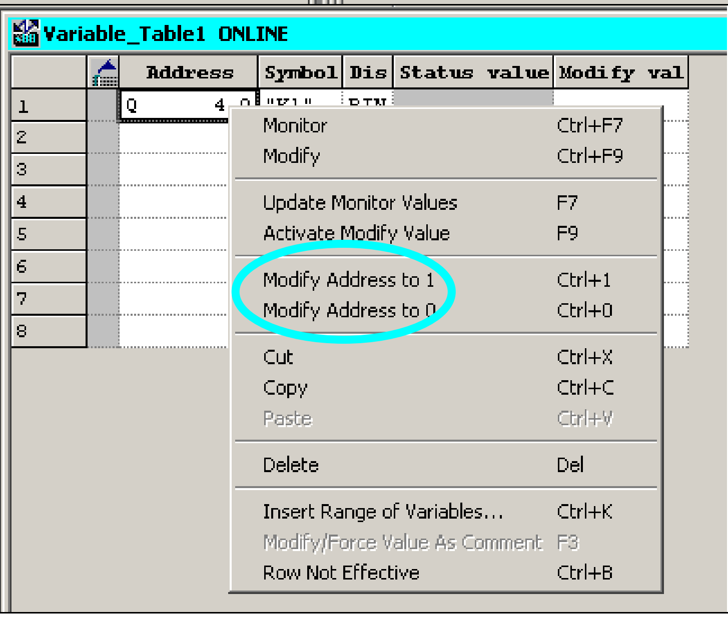
.





place the right mouse button in "Status value", then click "Modify Address to 1 or 0"
Now your actuator should react.

The actual Status is indicated in "Update Monitor Values".



Make sure that you now know all the output addresses which are used at your station.

e) Simulate the movements of your actuators in the right order according to the steps in your table on page 1.
Use "Modify address to 1" to activate the actuator.

f) Complete the Symbol Table.

Berufsschule für Fertigungstechnik	5. Analysing	PLC-Module

5.4 Electric Circuit

The Variable Table can be used to test the electric circuit. Create a Variable Table with the SIMATIC Manager for your outputs:

Task 5.4

a) Draw an electrical circuit of your station

IB0; OB4

24V

Input-address	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
	PLC							
Output-address	4.0	4.1	4.2	4.3	4.4	4.5	4.6	4.7

0 V

IB1; OB5

24V

Input-address	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
	PLC							
Output-address	5.0	5.1	5.2	5.3	5.4	5.5	5.6	5.7

0 V

Berufsschule für Fertigungstechnik	6. Sequence Chain	PLC-Module

6.1 Structured Program

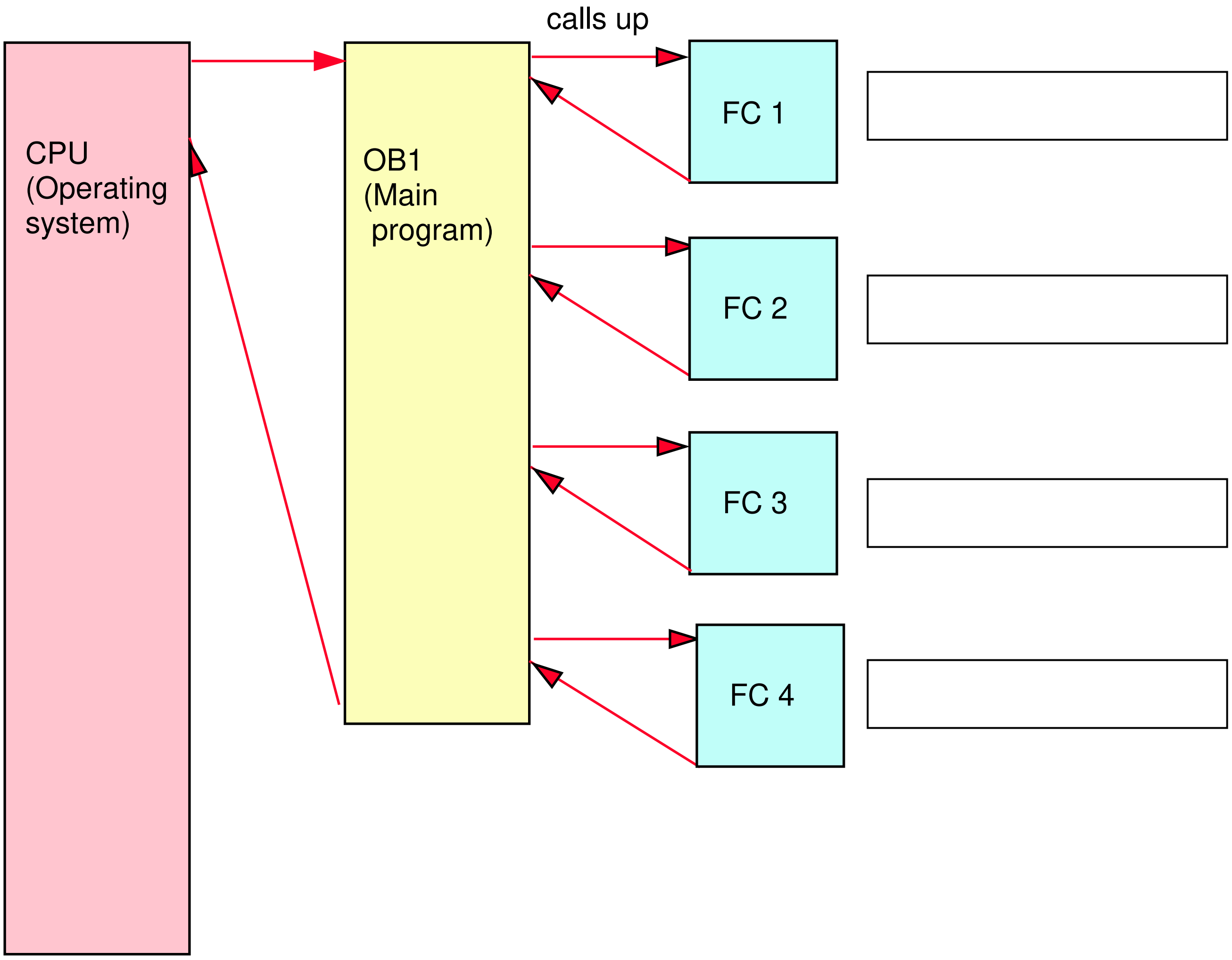
Programs are easier to handle if they have a structure, e.g.,

- Trouble shooting
- Extension
- Changing
- Copying

The structure of our program is organized into 4 functions (FCs). FC1: modes of operation, FC2: sequence chain, FC3: execution and FC4: indication

The CPU (Operating System) calls up OB1 as a standard.
In a structured program other blocks can be called up by the OB1: for example functions (FCs) or function blocks (FBs).
Note: If you forget to call up a FC in OB1 it will be ignored by the CPU.

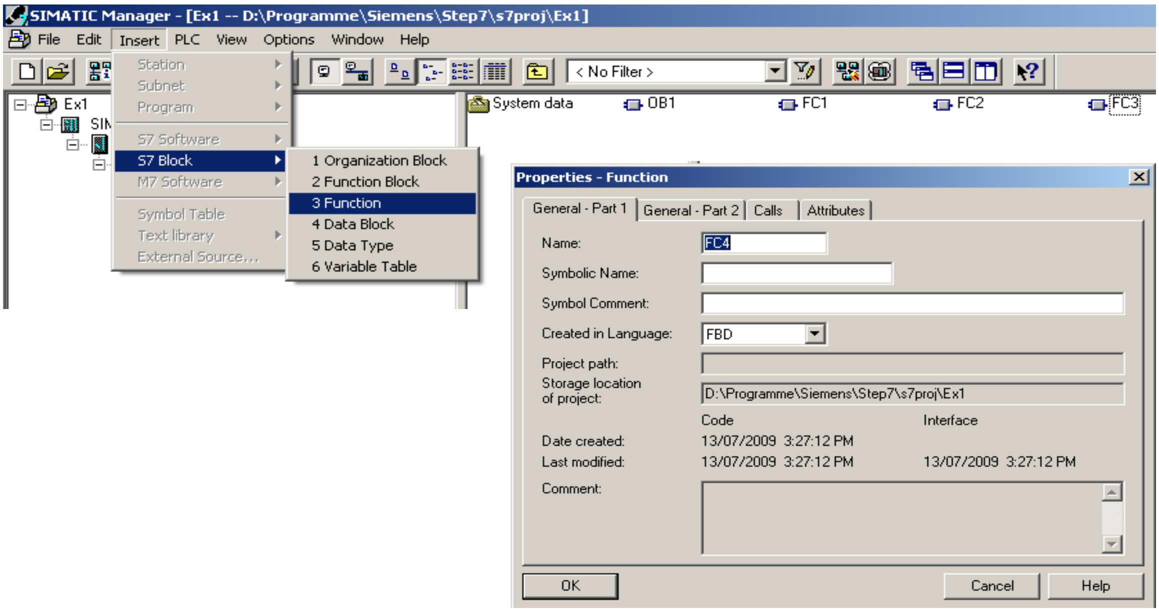
Please fill in the 4 functions of the FCs!



Structure your program:

Open your project Ex1.

Insert four S7 Blocks Function: FC1…FC4 in language FBD

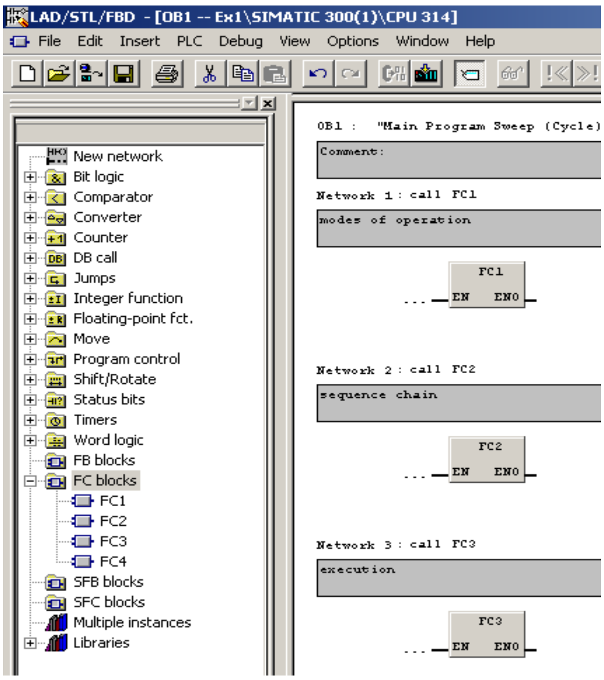


After this, the **4 FCs** must be called up by the OB1

Open OB1

Insert 4 Networks

and call up FC1…FC4:



Questions 6.1:

a) What are the reasons for structured programming?

b) Which structure do we use for our program?

c) Which part of the program is usually called up from the CPU?

d) What happens if you forget to call up FC2?

6.2 Sequence Chain

For a save operation of automatic processes the sequence chain is used as an industrial standard.

The basic principles are:

- **Only one**: There is only one step active at any time.
- **Memory Bit**: Every step of the sequence chain is set on a memory bit (not on an output Q). Step 1 = memory bit M1.0; step 2 = memory bit M2.0…
- **Set**: A step (e.g. a cylinder extends) is activated or set with the preceding step and a sensor (e.g. reed contact) makes sure that the step before is finished.
- **Reset**: the active step is deactivated by the next step or by the reset button.

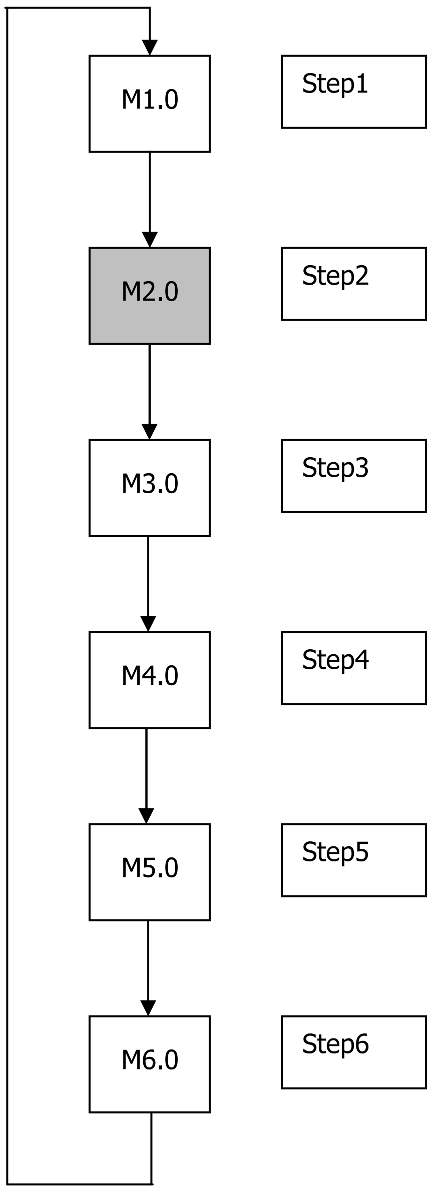
Note: no output is set in the sequence chain in FC2

Sequence chain in FC2

First step

If the station is switched on there is no step (memory bit) to prepare the first one.
The start button (1S0) and the sensors (1B1, 2B1), which indicate that the station is in initial position, are used as the condition for the first step. The reset conditons are the same in every other step.

a) Create the example network with FBD for step 1 in FC2.



Sequence chain in FC2

b) Create the example network in FBD for step 2 in FC2.

Execution of the sequence chain in FC3 with outputs:

The memory bits which are created by the sequence chain in FC2 are used to set and reset the outputs.

An output might be set in step 2 and reset in step 5 depending on how long the signal is necessary.

The output Q 4.1 should be set with step 2 and reset with step 5 or with the reset button I0.1.

c) Complete the network in FC3.

Questions on the text 6.2:

a) What are the basic principles for one step of the sequence chain in FC2?

b) How do you keep an output activated from step 1 to 4 in FC3?

c) How many steps of the sequence chain are active at the same time?

d) Why are the set conditions different in step1?

e) Sometimes 5/2 valves are used with one coil and spring return or with two coils. What difference does this make in your program in FC3?

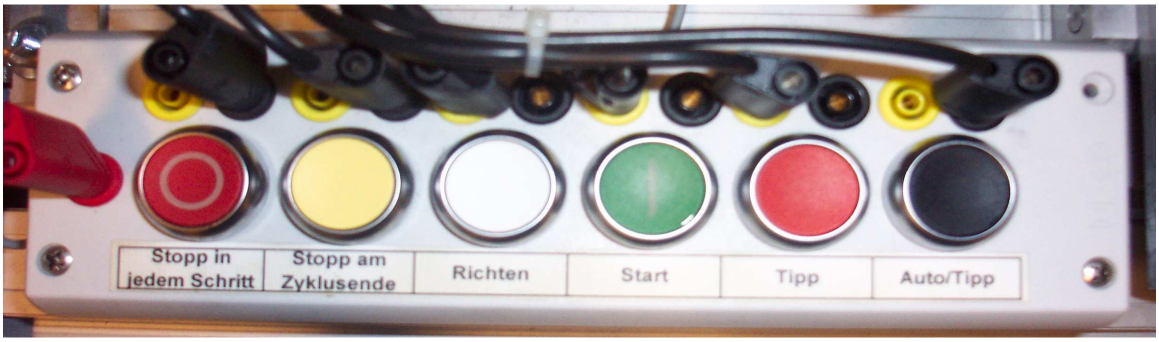
MOVE IT:

Program your station in FC2 and FC3 using the information given.

Note: The program will be extended with FC1 and FC4.

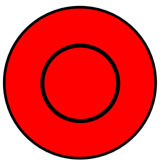
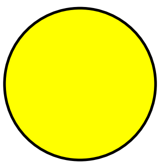
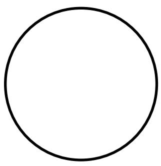
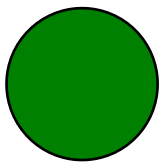
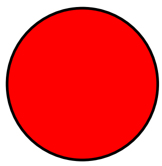
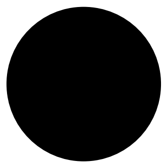
6.3 Pushbuttons and switches

Here is an example for a set of pushbuttons and switches which can be used in your project. Make sure that you have filled them in, in your Table of Symbols.



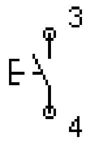
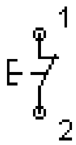
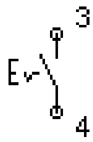
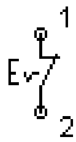
With our station, we use different types of “switches”:
Push buttons **pb**: give a signal for as long as you push them.
Switches **sw**: give a signal if you have pushed them once.
Both can be either normally open (**n.o.**) or normally closed (**n.c.**)

		address	type	function
S0	Start		pb n.o.	Starts the sequence chain
S1	Reset		pb n.o.	Resets all SR
S2	Auto/ Tipp		sw n.o.	If you are in Tip mode (S2 = 0) the next step is executed when S3 is pushed.
S3	Tipp		pb n.o.	If you are in Tipp mode (S2 = 0) the next step is executed when S3 is pushed.
S4	Stop			Stops the sequence chain in every step
S5	Finish		pb n.o.	Stops after a full cycle e.g. at the end of a work day

					
S4	S5	S1	S0	S3	S2
Stop	**Finish**	**Reset**	**Start**	**Tip**	**Auto/Tip**
Stops the sequence chain in every step	The station stops after the last step at the end of a cycle.	Resets all SR	Starts the sequence chain	If you are in Single Step (Tip) mode (S2 = 0) the next step is executed when S3 is pushed.	S2 = 1: Automatic mode S2 = 0 Tip (single step) mode

MOVE IT!

- Fill in the correct abbreviations for the symbols.

- Fill out all the pushbuttons and switches in your Symbol Table including the necessary information.

S7 Program(1) (Symbols) -- Ex1\SIMATIC 300(1)\CPU 314

	Status	Symbol	Address		Data type	Comment
1		S0	I	0.0	BOOL	Start pb n.o.
2		S1	I	0.1	BOOL	Reset pb n.o.

6.4 FC1 Modes of Operation

For our mechatronic system we use two modes of operation:

Automatic mode:
The system operates continuously without human input. The sequence chain works in a cycle without stopping.

It only stops:
- in case of an **emergency** (emergency stop)
- or after a full cycle at the end of a work day (finishing time).

Single step mode:
The next step of the sequence chain is executed when the single mode button is pressed. This mode is often used while:
- putting a station into operation
- or trouble shooting.

Conditions

In our structured programs we use FC1 for the modes of operation. The following conditions are necessary to allow your station to start:
- **no step is active** in the sequence chain
- the station is in **initial position** which is controlled by **sensors**

MOVE IT:
Note: The memory byte MB50 is chosen for networks in FC1.
Develop **4 networks** for FC1

Network 1: **No step active M50.0**:
Develop a network which makes sure that **no step of the sequence chain is active** and set **M50.0** if these conditions are true.

Network 2: **Initial position M50.1**:
Develop a network which makes sure that the **station is in its initial position** and set **M50.1** if these conditions are true.

Network 3: **Start** Memory bit **M50.2**:
Develop a network that sets a **Start Memory bit M50.2** if:
- M50.1 is true and
- M50.2 is true and
- if you press the start button **S0** (pushbutton n.o.)
- which is reset with the finishing time button **S5** (pushbutton n.o.)

Network 4: Single Step or Automatic Mode:
Develop a network for a **Mode of operation Memory bit M50.3** if:
- the **automatic** mode is switched on: **S2 = 1**
- the **single step** mode is on: **S2 = 0 and S3 = 1** (pushed).

6.5 FC4 indication

The indication usually shows the mode of operation of a station. We use three lamps with different colours:

- Green: Automatic mode, steady light
- Orange: Single step mode, blinks 1 Hz
- Red: an error has occurred, blinks 2 Hz.
  An error in this definition is if the sequence chain does not move for more than 10 seconds. If this happens we assume that there is something blocked or a loss of air pressure. The lamp indicates that some kind of maintenance is necessary.

MOVE IT:

Solve this indication with FC4 in your program.

Use the Help Function of the SIMATIC Manager and check **Clock Memory**

This action based training was developed within the Leonardo Da Vinci Transfer of Innovation Project:

"MODULES FOR VOCATIONAL EDUCATION AND TRAINING FOR COMPETENCES IN EUROPA"
"MOVET"
(PROJECTNUMBER **DE/10/LLP-LdV/TOI/147341)**

Module PLC

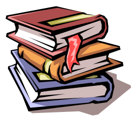
Solution

The aim of the training is to enable the apprentices to develop the skills, knowledge and competence for competence area 7 of the competence Matrix Mechatronics from the VQTS model (cf. Karin Luomi-Messerer & Jörg Markowitsch, Vienna 2006)

7.3 He/She can integrate and configure program-, control-, and regulation- mechanisms in mechatronic systems, program simple devices (in co-operation with developers) and simulate the program sequence before start-up.

Berufsschule für Fertigungstechnik	**1.Introduction PLC_SOL**	PLC-Module

1.1 History



Questions on the text 1.1:

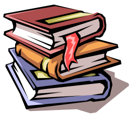
a) Why was the model change-over time consuming and expensive?
Because the complicated relay system was hard wired and needed to be rewired by skilled electricians.

b) What was the name of the first PLC and who is regarded as the father of it?

Modicon was the first PLC and the Dick Morley is regarded as the father of it.

Berufsschule für Fertigungstechnik	2. Modular PLC_SOL	PLC-Module

2.1 Hardware Configuration



Questions on the text 2.1:

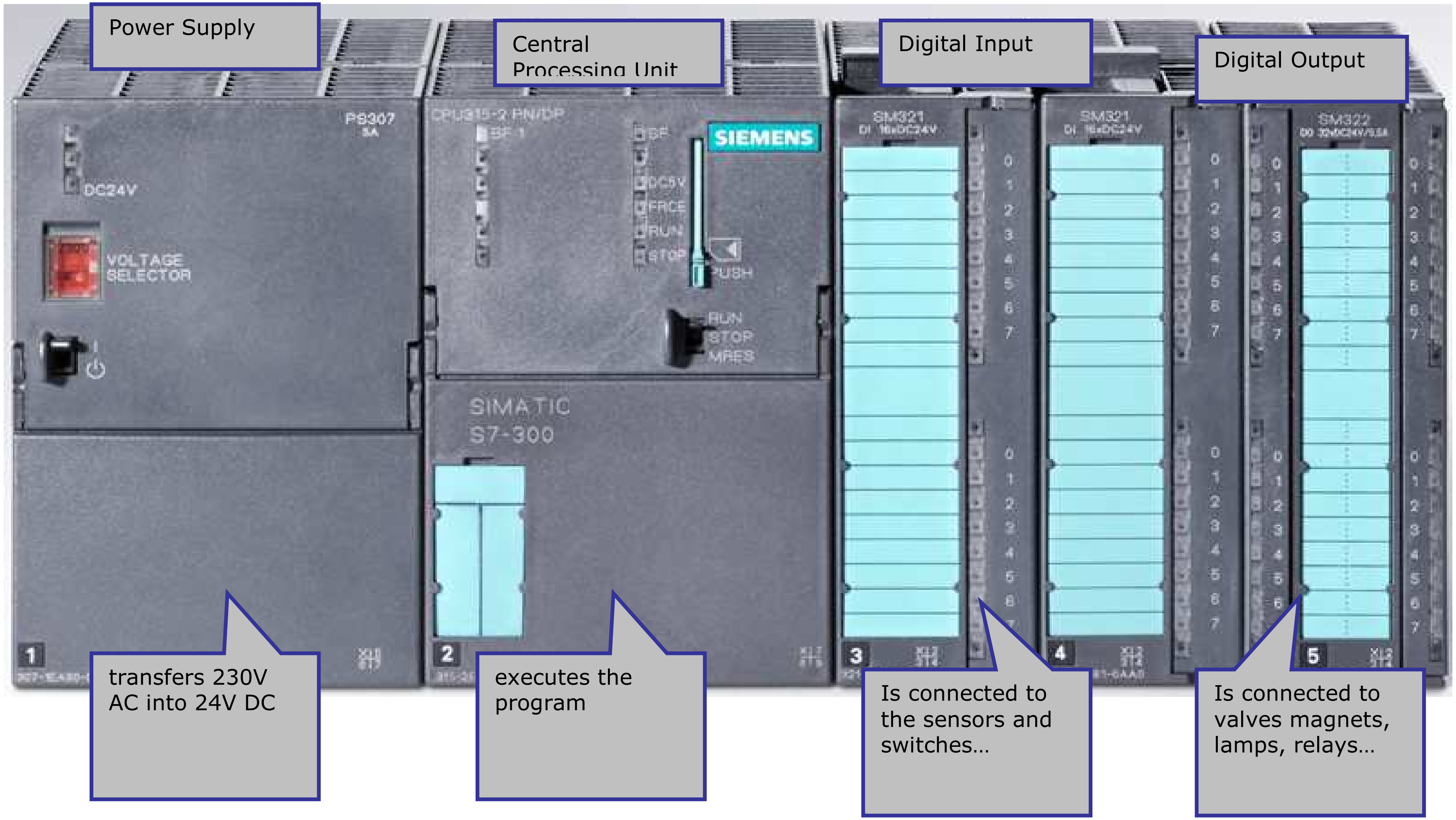
a) Explain the term “configuration”
The term "configuring" refers to the arranging of racks, modules, distributed I/O (DP) racks, and interface sub modules in a station window.

b) What does a configuration table represent?
Racks are represented by a configuration table that permits a specific number of modules to be inserted, just like a real rack.

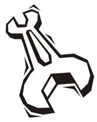
c) Which modules can be typically found in slot 1, 2, 3, 4, 5?
1: Power Supply (PS), 2: Central Processing Unit (CPU), 3: Interface Module (IM), 4/5: I/O-modules, communication processors (CP) or function modules (FM)

d) How many bytes are reserved for each slot?
4 bytes are reserved for each slot

2.3 Worksheet_Modular PLC Solution



2.5 Hardware Configuration



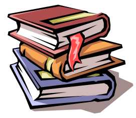
a) Which Input and Output addresses can be used with this project?
Please write them down:

Input addresses: I 0.0 … I 1.7

Output addresses: Q 4.0 … Q 5.7

Berufsschule für Fertigungstechnik	**3. Addressing SOL**	PLC-Module

3.1 Information Addressing



Tasks 3.1

a) Fill in the correct addresses in your worksheet from 2_Hardware Configuration.

b) What happens if you use the 16 bit DI module?
Every DI or DO module reserves 32 bits, so if you use a 16 bit module, the other 16 bits are lost.

c) Your colleague programmes an output address “Q 4.9”
Such an output address does not exist because there is no higher address than X.7

4.1 Basic Bit Logic

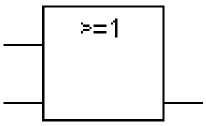
- Complete the function block symbols with in- and outputs.
- Complete the function table.
- Write a short description of the function.

Use the help function of the SIMATIC Manager to complete the function table:

Click the ? button and click on the Bit Logic Symbol to activate the Help or use F1.

Basic Bit logic: OR, AND, SR

>=1 : OR Logic Operation



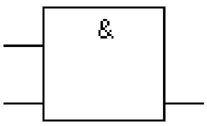
Function/truth Table

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Description

If the signal state of one of the addresses is 1, the condition is satisfied.

& : AND Logic Operation



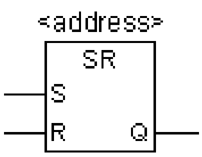
Function/truth Table

I1	I2	Q
0	0	0
0	1	0
1	0	0
1	1	1

Description

If the signal state of all operands is 1, the condition is satisfied.

SR : Set_Reset Flip Flop



Function/truth Table

I1	I2	Q
0	0	0
0	1	0
1	0	1
1	1	0

Description

Set reset Flip Flop is set when the signal state at input S is 1 and the signal state at input R is 0. If input S is 0 and input R is 1, the flip flop is reset. If the RLO at both inputs is 1 the flip flop is reset.

4.3 Information CPU Cycle

Questions on the text 4.3:

a) What can be found in a PII?
You will find the signal of an input high or low in the PII.

b) What can be found in a PIQ?
You will find the signal of an output high or low in the PIQ.

c) Describe the three steps of a CPU cycle?
- The status of the inputs will be stored in a PII.
- The processor executes the programme
- The results of the programme are written in the PIQ

d) Where is the programme that you wrote and how is it processed?
The programme is located in the programme memory. It will be processed step by step (successively).

e) Which other storage areas will be accessed during the CPU cycle?
Besides PII and PIQ, counters, timers and memory bits will be accessed during a CPU cycle.

4.4 Information CPU

Questions on the text 4.4

- Which modes are possible if you want to execute a program?

The modes are RUN or RUN-P

- Which mode do you have to select if you want to have writing access to the CPU for the PG and execute a program?

RUN-P

- Describe the status of the LEDs when a program is transferred from the PG to the CPU until it is active.

Program is transferring: STOP (orange), then RUN (green) blinks for a few seconds, then lights permanently.

- What is connected to the MPI-interface?

The programming device (PG), usually a computer.

- Describe the situation where a Memory Card is very useful.

The program can be saved without a battery in case of a power outage.

4.5 Using the Glossary

Task 4.5

a) Find out the following abbreviations:

PLC	*Programmble Logic Controller*
PII	*Process Image Input Table*
PIQ	*Process Image Output Table*
CPU	*Central Processing Unit*

4.6 Using Help Instructions

It is usually very helpful to read the description and study the example and the timing diagram.
Note: The timing diagram has two signal states 0 or 1.

Description

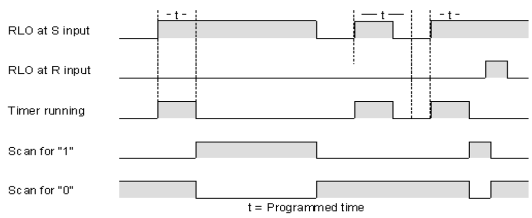
The **Assign On–Delay Timer Parameters and Start** instruction starts a specified timer if there is a rising edge (change in signal state from 0 to 1) at the Start (S) input. A signal change is always necessary to start a timer. The timer continues to run for the time specified at the Time Value (TV) input as long as the signal state at input S is 1. A signal state check for 1 at output Q produces a result of 1 when the time has elapsed without error and when the signal state at input S is still 1. When the signal state at input S changes from 1 to 0 while the timer is running, the timer is stopped. In this case, a signal state check for 1 at output Q always produces the result 0.

A change from 0 to 1 at the Reset (R) input of the timer while the timer is running resets the timer. This change also resets the time and the time base to zero. The timer is also reset if the signal state is 1 at the R input while the timer is not running.

The current time value can be scanned at the outputs BI and BCD. The time value at BI is in binary format; at BCD it is in binary coded decimal format.

Timing Diagram

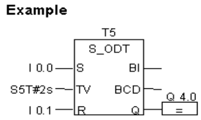
On-Delay timer characteristics:



Status Word

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
writes	-	-	-	-	-	X	X	X	1

Example



If the signal state of input I0.0 changes from 0 to 1 (rising edge in the RLO), timer T5 is started. If the specified time of two seconds (2s) elapses and the signal state of input I0.0 is still 1, the signal state of output Q4.0 is 1. If the signal state of input I0.0 changes from 1 to 0, the timer is stopped and output Q4.0 is 0. If the signal state of input I0.1 changes from 0 to 1 while the timer is running, the timer is restarted.

Start | Diplomarbeit | SIMATIC Manager - Ex 1 | LAD/STL/FBD - [FC1 -- Ü... | Dokument1 - Microsoft W... | Help on Function Block... | 16:44

<u>Task 4.6</u>
Study the example **Assign On-Delay Timer S_ODT** using the Help function and answer the following questions:

a) What is a rising edge?
The signal changes from 0 to 1
b) What is falling edge?
The signal changes from 1 to 0

Set your specified time of two seconds at TV (time value)
What happens at the output Q?
How long do you have a signal at Q …
c) if the signal at S is 8 seconds long?
6 seconds
d) if the signal on S is shorter than 2 seconds?
No signal at Q
e) if there is a signal at input R after 3 seconds?
The signal at Q lasts 1 second

Berufsschule für Fertigungstechnik	**5. Analysing SOL**	PLC-Module

5.1 Variable Table

Task 5.1

a) Which input bits from IB 0 have a 1-signal?

I 0.2 and I 0.5

b) What are the two main functions of the Variable Table?
The first function is to monitor inputs and the second function is to modify outputs.

5.2 Symbol Table

Question 5.2:

a) What is the difference between the Variable Table and the Symbol Table?
With the Variable Table, inputs can be monitored, and outputs modified. The Symbol Table is a documentation tool, which enables you to use these values for programming, and the comments from the Symbol Table also appear in the program when using the address.

Berufsschule für Fertigungstechnik	6. Sequence Chain SOL	PLC-Module

6.1 Structured Program

Questions 6.1:

a) What are the reasons for structured programming?

A structure makes it easier to change, copy, extend or troubleshoot a program.

b) Which structure do we use for our program?

Our structure is FC1: modes of operation, FC2: sequence chain, FC3: execution, FC4: indication

c) Which part of the program is normally called from the CPU?

OB1 is called by the CPU.

d) What happens if you forget to call FC2?

If FC2 is not called, it will be ignored by the CPU and therefore will not operate.

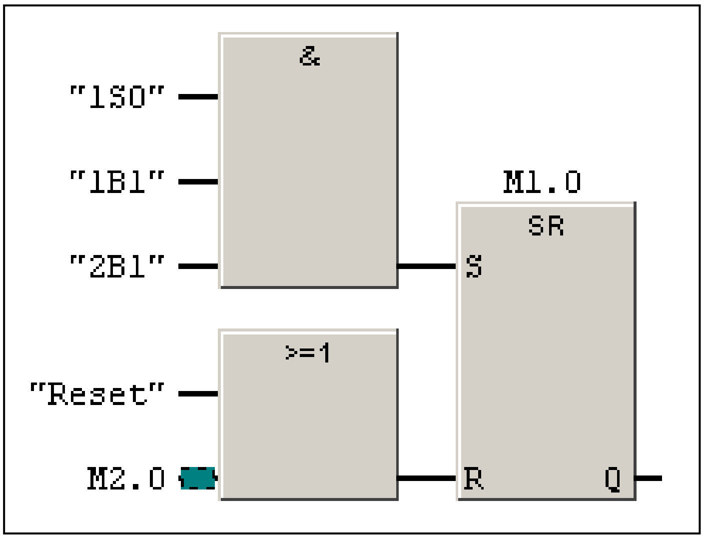
Berufsschule für Fertigungstechnik	6. Sequence Chain SOL	PLC-Module

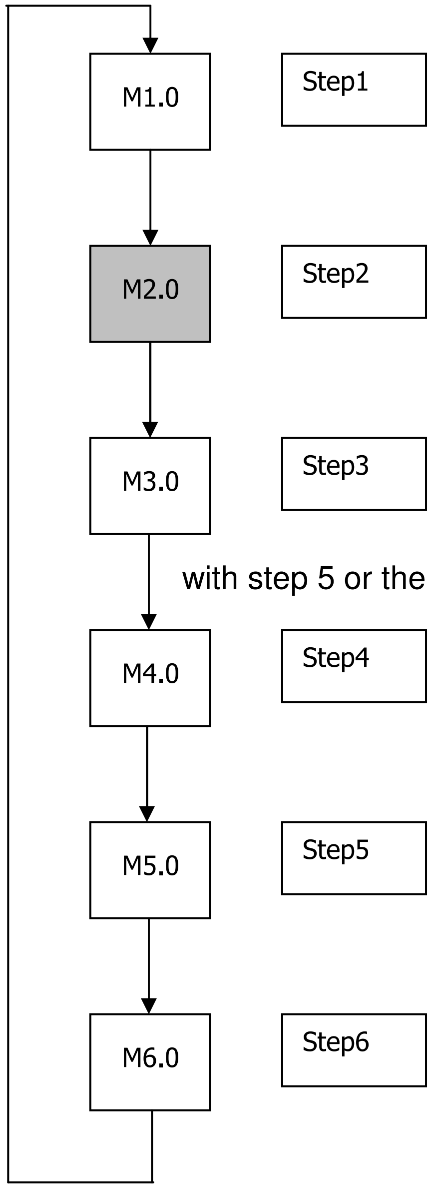
6.2 Sequence Chain

Sequence chain in FC2

First step

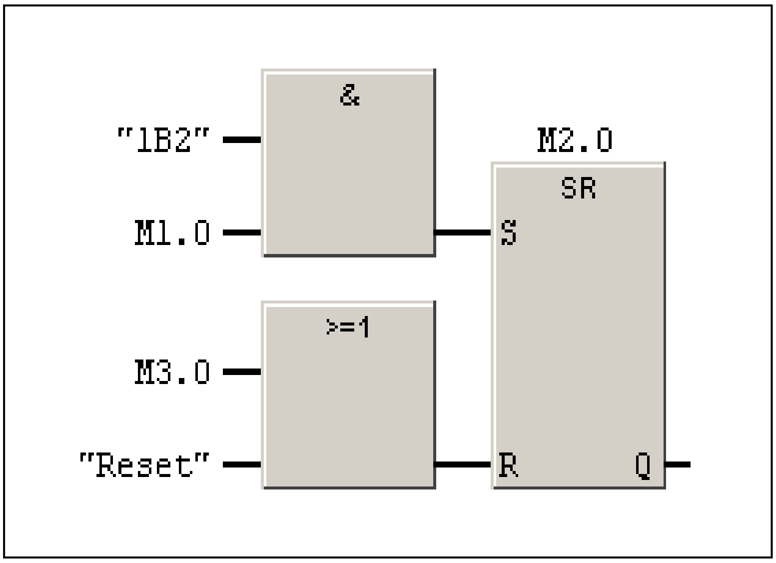
If the station is switched on there is no step (memory bit) to prepare the first one.
The start button (1S0) and the sensors (1B1, 2B1), which indicate that the station is in the initial position, are used as the condition for the first step. The reset conditions are the same in every other step.
Create the example network with FBD for step 1 in FC2.





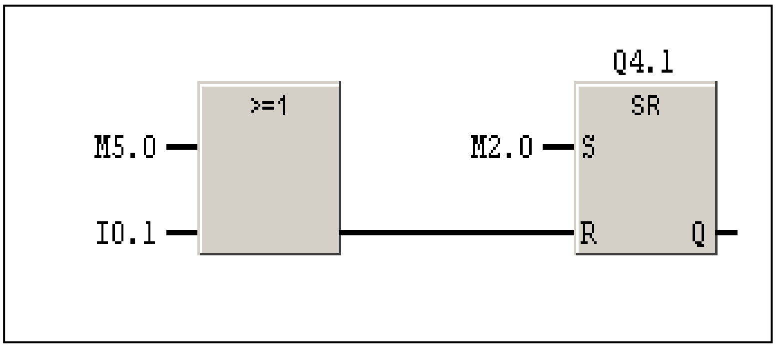
Sequence chain in FC2

Create the example network in FBD for step 2 in FC2.



<u>Execution of the sequence chain in FC3 with outputs:</u>
The memory bits which are created with the sequence chain in FC2 are used to set and reset the outputs.
An output might be set in step 2 and reset in step 5 depending on how long the signal is necessary.

The output Q 4.1 should be set with step 2 and reset with step 5 or the reset button I0.1. Complete the network with FBD in FC3.



Questions on the text 6.2:

a) What are the basic principles for one step of the sequence chain in FC2?

Only one*: There is only one step active at any time.*

Memory Bit*: Every step of the sequence chain is set on a memory bit.*

Set*: A step (e.g. a cylinder extends) is activated or set with the preceding step* *and* *a sensor (e.g. reed contact) makes sure that the preceding step is finished.*

Reset*: the active step is deactivated by the next step* *or* *by the reset button.*

b) How do you keep an output activated from step 1 to 4 in FC3?

Set it at step 1 with M1.0 and reset it with M4.0 at step 4.

c) How many steps of the sequence chain are active at the same time?

Only one.

d) Why are the set conditions different in step1?

Because when starting the station there is no preceding step as a set condition.

e) Sometimes 5/2 valves are used with one coil and spring return or with two coils. What difference does that make in your program in FC3?

5/2 way valves with spring return need a permanent signal such as from an SR; for valves with two coils, only a short impulse is necessary.

MOVE IT:

Program your station in FC2 and FC3 using the information given.

Note: The program will be extended with FC1 and FC4.

Berufsschule für Fertigungstechnik	**Glossary PLC**	PLC-Module

According to chapters

English	Deutsch
1. Introduction PLC	
Programmable Logic Control (PLC)	Speicherprogrammierbare Steuerung (SPS)
Connection programming control (CPC)	Verbindungsprogrammierbare Steuerung (VPS)
2. Modular PLC	
configure	konfigurieren
Central Processing Unit CPU	Rechnereinheit, Prozessor
slot	Hier: Steckplatz
rack	Hier: Profilschiene
Configuration table	Konfigurationstabelle
Module	Hier : Baugruppe
Programming device	Programmiergerät
Power Supply PS	Stromversorgung
Digital Input DI	Digitaleingang
Digital Output DO	Digitalausgang
3. Addressing	
Actuator (e.g. valve)	Aktor (z. B. Ventil)
Word (=2 Byte)	Wort (=2 Byte)
4. Programming	
Function Block Diagram (FBD)	Funktionsplan (FUP)
Function/truth table	Funktions-/Wahrheitstabelle
Process- Image Input Table (PII)	Prozessabbild der Eingänge (PAE)
Process-Image Output Table (PIQ)	Prozessabbild der Ausgänge (PAA)
Access	Zugang
Storage	Speicher
Counter	Zähler
Assign On-Delay Timer S_ODT	Einschaltverzögerung S_EVERZ
Rising edge	Steigende Flanke
Falling edge	Fallende Flanke
Value	Wert

5. Analysing	
Variable Table	Variablentabelle
to monitor	überwachen
to modify	ändern
Variable	Variable, Veränderliche
Symbol Table	Symboltabelle

6. Sequence chain	
Sequence chain	Schrittkette
Trouble shooting	Fehlersuche
extension	Erweiterung
execution	Ausführung
Indication	Anzeige
To call up	aufrufen
Memory bit	Merker bit
To extend	ausfahren
Reed contact	Reed Kontakt (magnetisch betätigter Schalter)
Initial position	Ausgangsposition
coil	Spule
Spring return	federrückstellung
pushbutton	Taster
switch	Schalter
Clock memory	Taktmerker

Berufsschule für Fertigungstechnik	**Paper and pencil test**	PLC-Module

Please answer the following questions/tasks.
Good luck!☺

1. Hardware configuration

Slot	Module	Order number	Firmware	MPI address	I address	Q address
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2	CPU 314	6ES7 314-1AE01-0AB0		2		
3						
4	DI16xDC24V	6ES7 321-1BH82-0AA0			0...1	
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				4...5

1.1 What are the main functions of the modules used in this hardware configuration? (4p)

____

____

____

____

1.2 Which in- and output addresses can be used in this project? (2p)

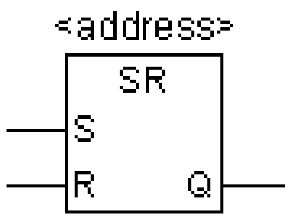
____

1.3 How many bytes are reserved for each slot? (1p)

____

2. Programming

2.1 S and R have a 1-signal. What signal do you get a output Q? (1p)



____

2.2 Describe the three steps of a CPU-cycle. (3p)

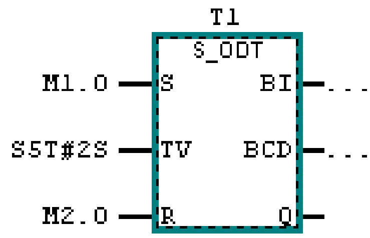
____

____

____

2.3 Which mode do you have to select on your CPU, if you want to transfer a program from the PG and execute it? (1p) ____________________

2.4 Under which conditions does output Q have a 1-Signal? (2p)



____

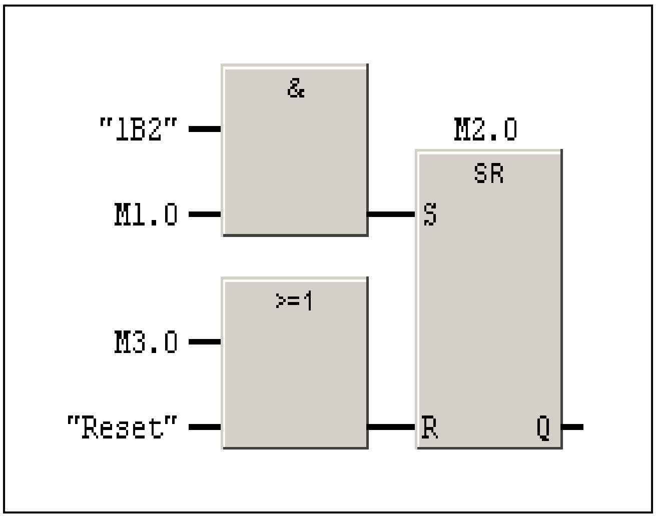
____

____

3. Analysing

3.1 What is the difference between a Variable Table and a Symbol Table? (4p)

3.2 The network shows step 2 of a sequence chain.
Step 2: When memory bit M2.0 is set, a cylinder moves out and gives a signal to 2B1.



Step 3: then the third step of the sequence chain should start. Draw the network for this step. (6p)

Σ 24

Berufsschule für Fertigungstechnik	**Paper and pencil test_SOL**	PLC-Module

Please answer the following questions/tasks.
Good luck!☺

1. Hardware configuration

Slot	Module	Order number	Firmware	MPI address	I address	Q address
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2	CPU 314	6ES7 314-1AE01-0AB0		2		
3						
4	DI16xDC24V	6ES7 321-1BH82-0AA0			0...1	
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				4...5

1.4 What are the main functions of the modules used in this hardware configuration? (4p)

PS: transforms 230 V AC to 24 V DC.

CPU: executes the program

DI: is connected to sensors and switches

DO: Is connected to valves magnets, lamps, relays

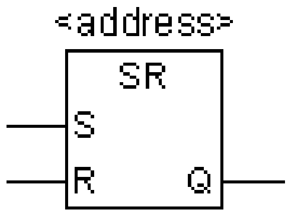
1.5 Which in- and output addresses can be used in this project? (2p)

IB0: I0.0..I0.7; IB1; I1.0…I1.7; QB 4: Q4.0…Q4.7; Q5.0…Q5.7

1.6 How many bytes are reserved for each slot? (1p)

4 Bytes are reserved for each slot

2. Programming



2.1 S and R have a 1-signal. What signal do you get a output Q? (1p)

The signal at Q is 0.

2.2 Describe the three steps of a CPU-cycle. (3p)

The status of the Inputsignals is transferred in the PII memory

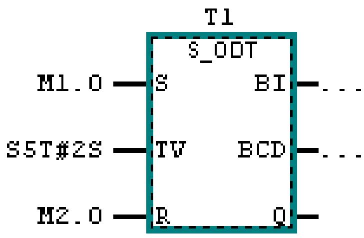
The program is processed

Status of the PIQ transferred to the outputs.

2.3 Which mode do you have to select on your CPU, if you want to transfer a program from the PG and execute it? (1p) *The mode is RUN-P*

2.4 Describe the conditions that output Q has a 1-Signal?

M1.0 has a 1-signal longer than 2 seconds and M2.0 has a 0-Signal.

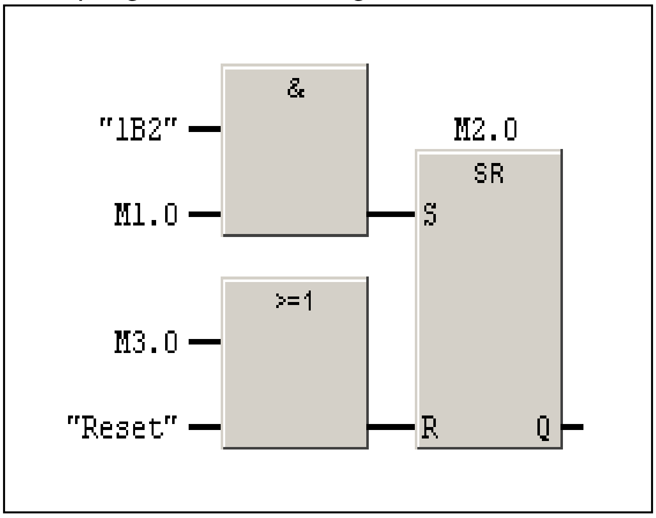


3. Analysing

3.1 What is the difference between a Variable Table and a Symbol Table? (4p)

With the Variable Table, inputs can be monitored, and outputs modified. The Symbol Table is a documentation tool, which enables you to use these values for programming, and the comments from the Symbol Table also appear in the program when using the address.

3.2 The network shows step 2 of a sequence chain.
Step 2: When memory bit M2.0 is set, a cylinder moves out and gives a signal to 2B1.



Step 3: then the third step of the sequence chain should start. Draw the network for this step. (6p)

The skills demonstration consists of three parts:

1. Work order **team** work (Monday to Wednesday morning)
2. Work order **single** work (Wednesday afternoon)
3. **Presentation** (Thursday morning)

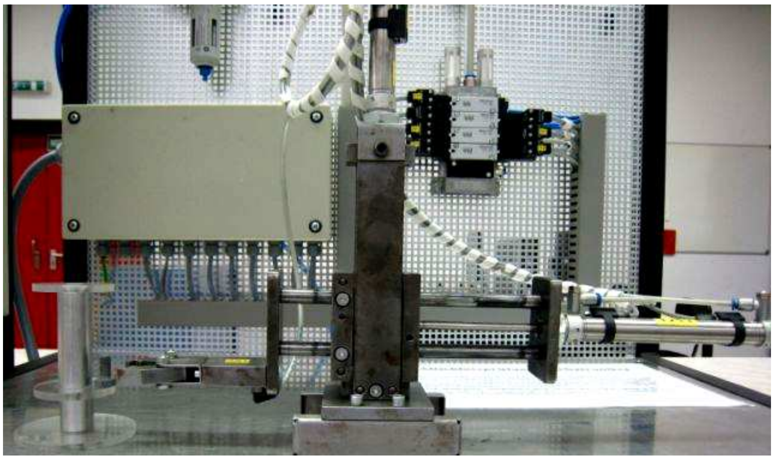
Good luck!

Work order team work

Please read the order carefully and program the station.

Basic programming in team work

The system consists of a **horizontal cylinder** (X-axis) a **vertical cylinder** (Z-axis) and a **gripper**.
The **initial position** you see in the photo with gripper **closed**. The initial position is indicated with a lamp.



Initial position

The mechatronic system has 2 operation modes:

- **Single step**
- **Automatic**

Both modes are switched on by two separate push buttons and shown by two separate lamps.

Operation mode **single step**:

Every movement of the mechatronic system can be carried out separately:
up and **down**; **fore**- and **backward**.
The movements are started with **4 pushbuttons**, one for every movement.
The front gripper is **opened** and **closed** also by **2 pushbuttons**.

Operation mode **automatic**

For starting the automatic mode, the mechatronic system must be in its initial position which can also be carried out by means of a pushbutton.
Function:
After the start button is pushed the gripper extends to its front position (X-axis) and picks a workpiece (bolt). Reaching the front position the gripper closes and retracts with the closed gripper holding the work piece.
Than it moves to the upper end position (Z-axis) extends to its front position and opens the gripper. After the work piece is placed the mechatronic system moves back in its initial position.

BMW Seidenader SWM BSFT	**Skills demonstration**	**Module PLC**

Work order single work

Please read the order carefully and program the station.

Additional Programming

- After every cycle in automatic mode the mechatronic system should stay for three seconds in its initial position.
- After 3 cycles the mechatronic system should remain in initial position and the lamp automatic mode should blink. The system is cleared by operating the push button automatic on and restarts after pushing the start button.

Please print out the following documents:

- Symbol Table
- Hardware configuration
- Program OB1, FC1…FC4

Questionnaire

1. <u>Presentation 5 minutes:</u>
(Hinweis : möglichst wenig Unterbrechung von Prüferseite, nur zur Aufmunterung/Unterstützung. Einer der Prüfer hält die ausgedruckten Dokumente in der Hand)

- Student should explain his project in general (1Min.):

Student: *My task was to programm…*
The program is structured in FCs…
There are two operation modes…

- Student presents the different operation modes: single step, automatic, addittional programming (4 Min.)

Student: *Every movement can be operated manually…*

2. <u>Discussion 5 minutes:</u>
(Vorschläge für den Prüfer)

- Hardwarekonfiguration: Could you explain your Hardwareconfiguration?
Which inputadresses can be used with your DI-Module?
What happens if you put the CPU in slot 3?

- FC1: How can you make sure that the system can´t be started while it is already running?

- FC2: Explain the structure of a typical step in FC2. Why did you structure it like this?

- FC3: When do you use equality function and when an SR-function in FC3

- FC4: How do make a blink signal?

Safety:
Press the emergency stop while the system is running: does the gripper loose the part?
Discuss different aspects of safety for the automatic system

BMW Seidenader MTU SWM BSFT	**Skills demonstration**	**Module PLC**
Students name:	Date:	

Report: Work order

Please tick off! (bitte abhaken!)

Operation mode **single step**:

- o Every movement of the mechatronic system can be carried out separately:
  up and **down**; **fore**- and **backward**.
- o The movements are started with **4** pushbuttons, one for every movement.
- o The front gripper is **opened** and **closed** also by **2 pushbuttons**.

Operation mode automatic

- o For starting the automatic mode, the mechatronic system must be in its initial position which can also be carried out by means of a pushbutton.

Function:

- o After the start button is pushed the gripper extends to its front position (X-axis) and picks a workpiece (bolt).
- o Reaching the front position the gripper closes and retracts with the closed gripper holding the work piece.
- o Than it moves to the upper end position (Z-axis) extends to its front position and opens the gripper. After the work piece is placed the mechatronic system moves back in its initial position.

Additional Programming

- o After every cycle in automatic mode the mechatronic system should stay for three seconds in its initial position.
- o After 3 cycles the mechatronic system should remain in initial position
- o and the lamp automatic mode should blink.
- o The system is cleared by operating the push button automatic on
- o and restarts after pushing the start button.

Please print out the following documents:

- o Symbol Table
- o Hardware configuration
- o Program OB1, FC1…FC4

Part 1: Presentation: 5 minutes

o good o non sufficient

Part 2: Discussion: 5 minutes

o good o non sufficient

company expert teacher

signature

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

Certificate

Module PLC

Ms/Mr *first name last name*

born *dd.mm.yyyy*

has successfully taken part in

90 hours of PLC (Programmable Logic Control) training

at

Städtische Berufsschule für Fertigungstechnik München

and

SWM GmbH

from 3rd of October 2011 to 21st of October 2011.

He has programmed a mechatronic system with PLC Siemens S7-300 in teamwork. Additionally he has successfully programmed various functions in single work. He has passed the final test (paper and pencil test; skills demonstration and technical discussion) successfully. All communication during the training and the team work was in English language.

Herewith we certify that

he can integrate and configure program-, control-, and regulation mechanisms in mechatronic systems, program simple devices (in co-operation with developers) and simulate the program sequence before start-up.

(Competence level description 7.3 according to VQTS model)

These Learning Outcomes are associated to EQF Level 4.

Munich, 21st of October 2011

Bernhard Hanslmaier
Team leader technical apprenticeship

Friedrich Dreßl
Städtische Berufsschule für Fertigungstechnik
Headmaster

BSFT

Imprint

This module was developed

at

Städtische Berufsschule für Fertigungstechnik (BSFT) München
Deroystr. 1
80335 München
Germany

www.ft-deroy.musin.de
phone.: +49 89 233 355 98

from

Manfred Schauhuber
Stephanie Bock
Arthur Joretzki

contact:
Manfred.schauhuber@bsz-deroy.muenchen.musin.de