International Journal
Information Technology Applications (ITA)

Volume 14, Number 2, December 2025

CONTROL AND VISUALIZATION SYSTEM FOR A LABORATORY MANIPULATOR MODEL

Marián Choma, Ján Ulej, Ján Cigánek, Hortenzia Wolentová

Abstract:

This paper presents the design and implementation of a comprehensive control and visualization system for a laboratory model of a vacuum manipulator manufactured by Fischertechnik. The main objective of this work was to create an automated workstation using industrial standards. In the first part, the article analyzes the model's hardware capabilities, including actuators and sensors. It then describes the hardware configuration in the TIA Portal environment, where the Siemens S7-1200 industrial PLC plays a key role. The creation of the control algorithm in Ladder Diagram language, the processing of high-speed signals from encoders, and the implementation of statistical functions are discussed in detail. The solution also includes the design of a graphical user interface for the HMI panel, which enables real-time monitoring and diagnostics. The functionality of the entire system was verified using the "Tower of Hanoi" task.

Keywords:

SCADA, HMI, PLC, manipulator, control system, visualization system, Tower of Hanoi.

Introduction

Efforts to streamline the handling of objects have led from simple mechanical tools to the development of complex robotic systems.

A key requirement for modern manipulators is the ability to perform precise tasks autonomously and repeatedly, which necessitates the use of powerful control units. In industrial practice, this role is fulfilled by programmable logic controllers (PLCs), which ensure the necessary stability, robustness, and flexibility in the control of technological processes.

The main objective of this article is to present the design and implementation of a control system for a laboratory model of a Fischertechnik vacuum manipulator using a SIMATIC S7-1200 PLC. The entire process, from hardware configuration to the creation of control logic in the LD (Ladder Diagram) language, was focused on achieving high positioning accuracy of the arm. The functionality of the solution was verified using the "Tower of Hanoi" test scenario, which tested the system's ability to process fast signals from magnetic encoders in real time.

An important part of the design is a visualization application created in WinCC Basic, which provides the operator with feedback and enables component diagnostics based on statistical operational data.

The article documents the key technical steps of the implementation, describes configuration adjustments for processing high-frequency pulses, and evaluates the achieved results in terms of control effectiveness.

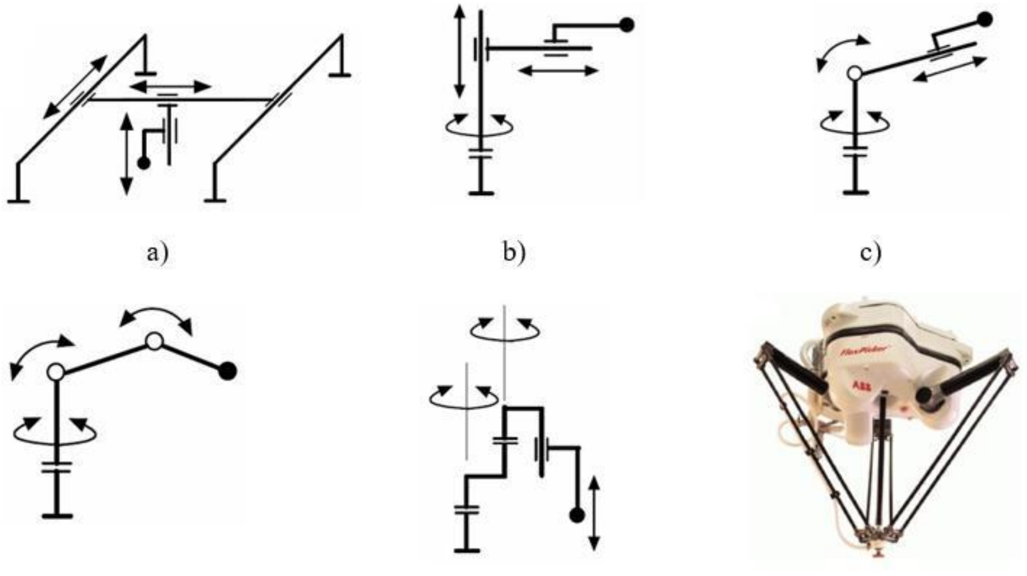
1 Introduction to Manipulators

A robotic arm, often referred to as a manipulator, is a fundamental component of an industrial robot. Its role is to manipulate objects in space, which includes gripping, moving, and precisely positioning them. From a structural standpoint, the manipulator consists of an arm (which provides the main translational movements), a wrist (used for precise rotation of the tool), and an end-effector, which can be a gripper, a suction cup, or a welding head [1].

A. Classification of Mechatronic Arms

The names of the arms are usually derived from their kinematic structure. To properly understand the control system, it is important to distinguish between the basic types [2]:

- **Cartesian structure (Fig.1a):** Motion occurs along three mutually perpendicular axes (X, Y, Z). The advantage is a simple kinematic model; the disadvantage is the robot's large size relative to the workspace.
- **Cylindrical structure (Fig.1b):** Combines one rotational and two linear movements. It allows access to recesses and open spaces. Our laboratory model uses this very structure.
- **Spherical (polar) structure (Fig.1c):** Contains two rotational and one linear joint.
- **Articulated (anthropomorphic) structure (Fig.1d):** Most commonly used in industry, resembles a human hand, and has 5 or more degrees of freedom [3].
- **SCARA structure (Fig.1e):** Designed for fast and precise horizontal movement, often used in assembly lines [3].
- **Parallel structures (Fig.1f):** They excel in high dynamics, rigidity, and often load-bearing capacity. However, their limited workspace and more complex design can be problematic.



a) b) c)

d) e) f)

Fig.1. Structures of mechatronic manipulators [3].

B. Types of Drives in Mechatronics

The choice of drive directly influences the design of the control algorithm. In mechatronic systems, there are three basic types [4]:

1. **Pneumatic actuators**: These use compressed air. They are simple and reliable, but have lower positioning accuracy. In our model, this actuator is used for the end effector.
2. **Hydraulic drives:** These use fluid and are designed to handle very heavy loads.
3. **Electric actuators**: The most common type in robotics. These include stepper motors for precise positioning without feedback, and servo motors, which use a closed-loop control system with encoders for maximum precision [5].

2 Control Systems

In mechatronics, control is understood as a goal-oriented process whose purpose is to achieve the desired state or behavior of a system (the controlled object). This process involves providing inputs to the system in such a way that the desired output is achieved, while taking into account criteria such as quality, time, and cost [6].

A. Principles of Automated Control

Automated control is a process in which automated systems are used to monitor and influence a process without direct human intervention. The automated control process consists of four basic steps [7]:

1. **Measurement:** Collecting data on the process status using sensors.
2. **Processing:** Analysis and evaluation of measured data by the control unit.
3. **Control**: Adjusting the control action to achieve the desired result.
4. **Monitoring**: Monitoring the process and adjusting control as needed based on current conditions.

In terms of feedback, we distinguish between control (open-loop without feedback) and regulation (closed-loop with feedback), where the system continuously evaluates the deviation from the target value [8].

B. Types of Control Algorithms

Automatic control can be implemented in various ways, which differ depending on how the control system intervenes in the controlled process. We classify it into several types [6]:

- **Logic control**: Works with binary signals (states 0 and 1). An example is turning a motor on or off based on the activation of a limit switch.
- **Continuous control:** deals with variables that change continuously over time and are defined at every point in time.
- **Discrete control:** Processes inputs and outputs as a sequence of values obtained at regular time intervals. These intervals determine the so-called sampling period. No measurement or output adjustment takes place between individual samples.
- **Fuzzy control:** Suitable for systems that are difficult to describe mathematically. It uses qualitative values (e.g., "low," "medium," "high") instead of precise numbers [9].

C. Microprocessor Platforms (Arduino and Raspberry Pi)

Control systems are an integral part of modern technology, where they ensure the automation and efficient management of various processes. Among the most commonly used systems, which have found application not only in education but also in practice, are:

Arduino is an open-source platform for easily creating electronic projects, combining hardware in the form of a programmable microcontroller board with the Arduino IDE software development environment. Programs are written in a language similar to C++ and then uploaded directly and conveniently to the board via a USB cable, which significantly simplifies the entire development process [11]. One of the most commonly used boards for beginners in the field of microcontrollers is the Arduino UNO.

The Raspberry Pi is an affordable, credit-card-sized single-board computer developed to make computer science and programming education more accessible. When connected to standard peripherals, it functions as a full-fledged, energy-efficient personal computer powered by a Broadcom chip featuring an ARM processor and a graphics unit. The system runs primarily on the official Raspberry Pi OS, which is optimized for easy programming in Python but also supports other languages such as C++, Java, and JavaScript without issue, and is available on the market in several hardware versions. The Raspberry Pi Zero is one of the smallest and most energy-efficient versions of this platform. It is based on the original Raspberry Pi 1 Model B [12].

D. Industrial PLCs (Programmable Logic Controller)

For industrial applications requiring high reliability in dusty and vibrating environments, a PLC is the only option [9]. A PLC operates in a continuous cycle (Fig.2):

1. Reading the status of the inputs.
2. Execution of the control program.
3. Update the status of outputs [13].

Unlike Arduino, PLCs (such as the Siemens S7-1200) are designed for real-time operation and direct communication with industrial sensors and SCADA systems.

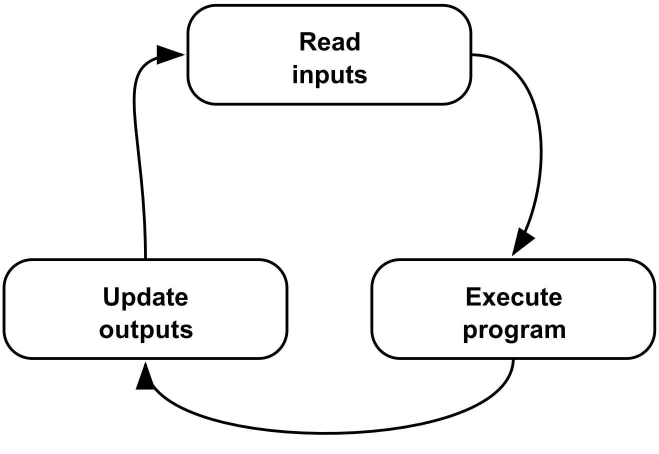


Fig.2. Simplified PLC cycle.

The programmable logic controller (PLC) market is dominated by a few companies. Among the best-known are, for example [14]:

- **Siemens:** A German technology giant that is among the leaders in the field of automation systems. Its SIMATIC series of PLC devices is known for its modularity, durability, and flexibility.
- **Allen-Bradley:** A division of Rockwell Automation that specializes in developing advanced manufacturing control solutions, primarily for the North American market.
- **Mitsubishi Electric:** A Japanese manufacturer offering a wide range of automation technologies, including control systems and industrial robots.
- **Schneider Electric:** A French company with many years of experience in energy management and industrial automation.
- **Omron:** Another Japanese company that focuses on developing advanced automation solutions, including sensors and control units.

E. Process Visualization

Visualization is used to graphically display production processes in real time. It allows operators to monitor machine status, material flows, and alarms, thereby simplifying the management and optimization of operations. Two main approaches are used for this purpose: comprehensive SCADA systems and HMI operator panels.

SCADA (Supervisory Control and Data Acquisition) is a comprehensive hardware and software system designed for centralized monitoring, data collection, and control of a large number of remote devices (pumps, motors, production lines). It is used in complex or hazardous processes that need to be automated.

Key SCADA Components:

- **Process equipment:** Sensors (measure temperature, level) and actuators (perform actions, e.g., open valves).
- **Data acquisition and control units:** Local devices that communicate with a central system. These include RTUs (for harsh, remote environments), PLCs (for industrial facilities), and IEDs (intelligent devices, often used in the energy sector).
- **MTU (Master Terminal Unit):** The main "brain" and central software of the SCADA system, which collects data and sends commands.
- **Communication interface:** A network (wired or wireless) connecting all system components [15].

HMI (Human-Machine Interface) is a tool (hardware + software) through which an operator communicates with a system (such as a touchscreen panel on a machine or a PC application). It allows the operator to monitor the status of the equipment and directly control it (change settings, start processes) [16].

The HMI consists of:

- **Hardware**: Touchscreens, command buttons, and displays or indicator lights.
- **Software:** A graphical interface with buttons, graphs, and alarms. The software also stores data and generates reports.

There are many manufacturers of this software on the market; among the best known are Rockwell Automation (FactoryTalk View), Siemens (WinCC), and the Czech company Microsys (Promotic) [16].

3 Workstation Implementation and Hardware Configuration

This chapter provides a detailed description of a specific laboratory model and the method for integrating it with the Siemens control system.

A. Technical Specifications of the Manipulator Model

The robotic manipulator from Fischertechnik (Fig.3) is a mechatronic system with three degrees of freedom. As mentioned in the theoretical section, it consists of a cylindrical structure in which two axes provide linear motion (vertical and horizontal) and one axis performs rotational motion. Linear movements are achieved using precision threaded rods that convert the motor's rotational motion into translational motion. The rotational motion of the base is provided by a worm gear, which ensures self-locking and high torque [17].

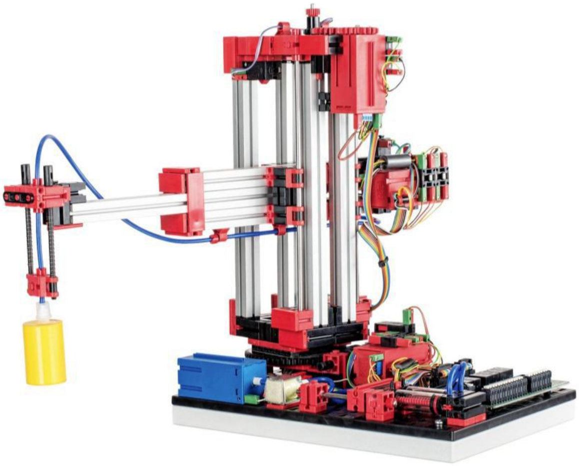


Fig.3. 24V robotic manipulator [17].

All three axes of motion are driven by permanent magnet DC motors. Their technical parameters are key to the design of the control loop:

- **Rated voltage:** 24 V DC.
- **Performance specifications:** Maximum power 2.03 W at 214 rpm.
- **Gear mechanism**: Each motor has an integrated gearbox with a 25:1 gear ratio.

For precise positioning, the motors are equipped with magnetic encoders (Hall sensors). The sensor generates 3 pulses per motor revolution. Given the gear ratio, the system actually generates 75 pulses per revolution of the output shaft. This value was used in the control algorithm to calculate the arm's trajectory in millimeters or degrees of rotation [18, 19].

The model uses a vacuum system for handling objects, which is the preferred method in industry for handling light and flat items.

The system consists of (Fig.4):

- **Diaphragm pump:** Functions as a compressor, operates at 24 V, and generates a pressure of approximately 0.7 bar. Its current consumption is 70 mA, which allows for direct power supply from the PLC output modules while maintaining safety margins [10].
- **Solenoid valve:** A 3/2-way valve controls the vacuum supply to the suction cup. When de-energized, the suction cup is vented to atmospheric pressure, ensuring immediate release of the object without residual vacuum.

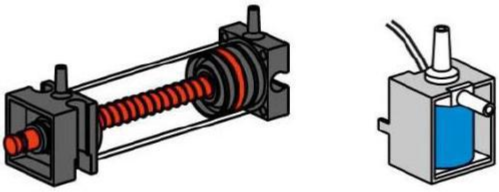


Fig.4. Pneumatic cylinder and 3/2-way solenoid valve [10].

The entire laboratory model was mounted on a mounting plate measuring 400 × 500 mm. Key control components are mounted on a DIN rail beside the model (Fig.5):

1. **PLC Siemens S7-1200:** Central Control Unit.
2. **PM1207 Power Supply:** Provides a stable 24 V supply for both the PLC and the model itself.
3. **WAGO terminal blocks:** Used to neatly connect the 26-pin manipulator connector to the PLC's I/O terminals.

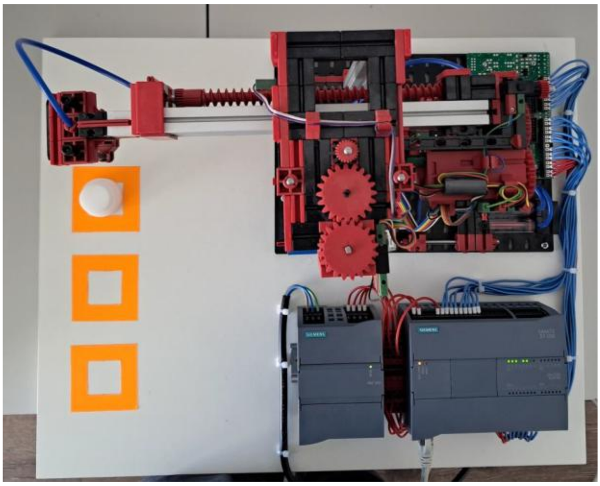


Fig.5. Complete manipulator model

The connection was made using CYA 1×0.5 single-core conductors, which were terminated with ferrules to ensure a high-quality electrical connection and long-term system reliability.

To verify the functionality of the robotic manipulator's control system and to visualize the controlled process itself, we chose the well-known Tower of Hanoi puzzle as a test case.

Three square positions were marked on the base (Fig.17), with three cylinders of different diameters placed on the first one, arranged from largest to smallest. The manipulator's task was to move all three cylinders from the initial position to the last marked position while adhering to the following rules:

- Only one cylinder can be moved at a time.
- This step involves removing the top cylinder and placing it on top of the cylinder in a different position.
- It is prohibited to place a larger cylinder on top of a smaller one.

B. SIMATIC S7-1200

The SIMATIC S7-1200 (Fig.6) control unit from Siemens is a compact yet powerful solution for automating small to medium-sized applications. It offers an excellent price-performance ratio, a robust design suitable for industrial environments, and easy installation on a standard DIN rail.

The configuration, programming, and diagnostics of this PLC are performed using the TIA Portal development environment, which significantly streamlines development and reduces the time required to commission the system.

Key Features and Functions:

- **Integrated I/O:** The base unit features 14 digital inputs, 10 digital outputs, and 2 analog inputs. The status of each digital signal is indicated by an LED, which facilitates visual monitoring and diagnostics.
- **Modularity:** The system is fully expandable. Additional I/O or communication modules can be flexibly added to the base unit according to the specific requirements of the application.
- **Advanced features:** The PLC natively supports technological features such as high-speed counters (HSC), pulse-width modulation (PWM), and speed and position control.

Fig.6. SIMATIC S7-1200 [10].

C. TIA Portal

The central software platform for the design, configuration, and programming of the control system in this project is the TIA Portal (Totally Integrated Automation Portal) from Siemens. It provides a unified development environment that combines program creation, hardware configuration, testing, simulation, and diagnostics into a single integrated solution. This approach eliminates the need for external tools, thereby significantly optimizing the development cycle and reducing the risk of integration errors. An overview of environment is shown in (Fig.7).

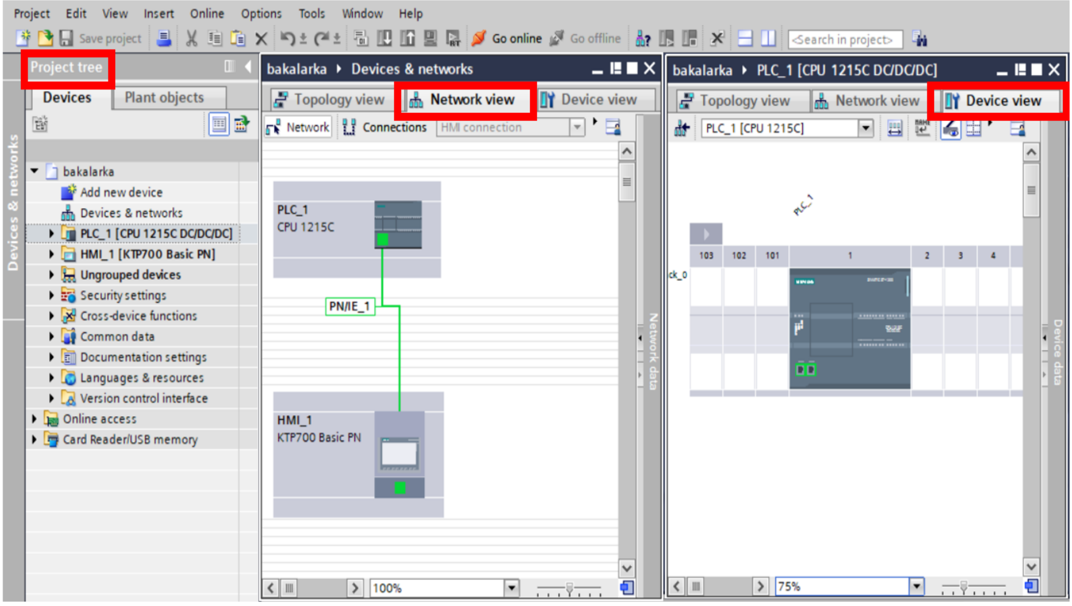


Fig.7. Main areas of the TIA Portal work environment.

The platform fully supports the programming of PLC devices (including the S7-1200, S7-1500, S7-300, and S7-400 families), as well as the configuration of HMI panels, frequency converters, and other network components. A significant benefit is its advanced simulation tools, which enable detailed validation of control logic even before deployment on actual hardware.

The workspace is optimized for clarity and is divided into three key areas:

- **Project View:** Provides a structured tree view of all files, program blocks, variables, and project settings in one place.
- **Device View:** Used for the graphical configuration of the PLC's hardware structure, its expansion modules, and the parameterization of I/O addresses.
- **Network View:** Displays the network topology and allows for easy configuration of communication links between individual nodes, such as PLCs and HMI panels.

The LD (Ladder Diagram) language was chosen for the development of the control algorithm itself. It is a proven industrial standard that is preferred for its visual clarity and the ease of implementing functional blocks (such as timers and counters).

The LD program is structured into separate logical networks. Graphically, it resembles an electrical circuit diagram:

- The signal flows from left (relative to the imaginary power phase) to right.
- Evaluation occurs through a series of contacts (representing physical inputs or logical conditions linked to BOOL-type variables).
- If all logical conditions in a given branch are met (the variables evaluate to TRUE), the signal reaches the end of the network and is sent to the coil, which then activates the corresponding output or sets an auxiliary variable.

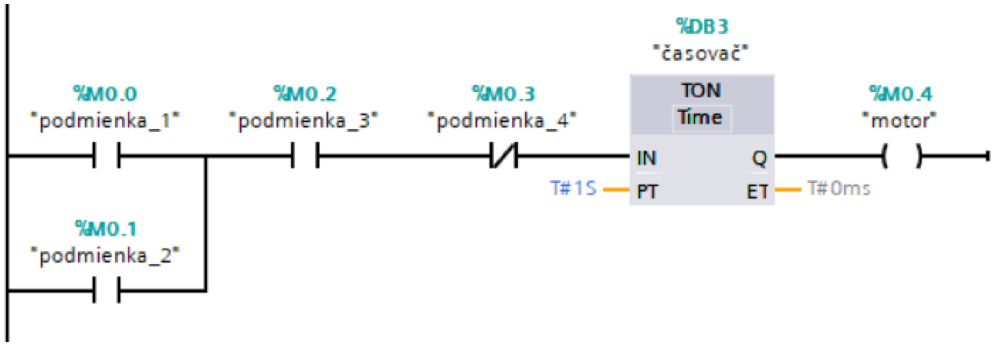


Fig.8. Sample program in the LD language in the TIA Portal environment:
condition 1 2 3 4, timer and motor.

D. WinCC Basic

WinCC Basic is a development software tool fully integrated into the TIA Portal platform, primarily designed for the design and configuration of SIMATIC Basic Panel operator panels. Its native integration ensures seamless connectivity between HMI (Human-Machine Interface) solutions and control systems such as the SIMATIC S7-1200 or S7-1500 PLCs..

The primary purpose of the system is to develop an intuitive graphical user interface through which the operator monitors the status of technological processes, intervenes in control operations, and visually evaluates operational data.

Key features and capabilities of the environment:

- **Hardware and network configuration:** Configuring HMI panel parameters and establishing secure communication with the master PLC.
- **Visual design:** Creating user interfaces, defining logical navigation, and implementing interactive elements (control buttons, input/output fields, status indicators).
- **Data and variable management:** Creating local HMI variables (tags) and mapping them directly to variables in the PLC program structures.
- **Alarm management:** Configuration of alerts and diagnostic messages designed to immediately notify the operator of malfunctions or critical changes in the process.
- **Integrated simulation:** Tools for compiling and performing full-scale software simulation of HMI projects directly on a PC.

E. HMI Panel KTP700 Basic

The SIMATIC HMI KTP700 Basic from Siemens is used for the visualization of operational data and the interactive control of technological processes. It is configured natively in the TIA Portal (WinCC Basic) environment, ensuring seamless integration with SIMATIC PLC systems (e.g., S7-1200 and S7-1500). Key features include:

- **Display:** 7-inch color TFT display (800 × 480 pixel resolution) with support for multilingual graphical interfaces.
- **Controls:** A combination of a touchscreen for intuitive navigation and hardware buttons for quick access to selected functions.
- **Applications:** Versatile use in industrial automation, construction, and the energy sector.

4 PLC Control Algorithm

The control program was developed in the TIA Portal v17 development environment. A Siemens S7-1200 PLC (CPU 1215C DC/DC/DC, part number 6ES7 215-1AG40-0XB0) was selected as the central control unit. Network communication between the development PC and the PLC took place via the Profinet (PN/IE) interface, with the PLC assigned the static IP address 192.168.0.234.

A. Variable Structure (PLC Tags)

To maintain code clarity and facilitate troubleshooting, the variables (tags) have been logically organized into four separate tables:

- **Inputs:** Mapping of physical input signals from sensors and buttons.
- **Outputs:** Control variables for physical actuators (motors, compressor).
- **Memory (Auxiliary variables):** Used for transitions in state machines, logical conditions, and temporary memory states.
- **Visualization:** Variables reserved exclusively for communication with the HMI panel.

B. Configuration of input filters for encoders

The laboratory manipulator uses motors with encoders to measure position, which generate pulses at a frequency of up to 1 kHz. The standard value of the PLC’s digital input filter (6.4 ms) was insufficient for this application. To enable the control unit to reliably detect short pulses, the hardware filters for the relevant inputs (DI 14/DQ 10) were reduced to a minimum value of 0.1 μs.

C. Program Architecture and Motion Control

The main task of the control algorithm is to ensure the sequential movement of three cylinders (labeled 1, 2, and 3) between working positions (labeled A, B, and C) (Fig.9).To ensure clarity, and facilitate testing, the entire program was divided into 18 logical sections (Fig.10).

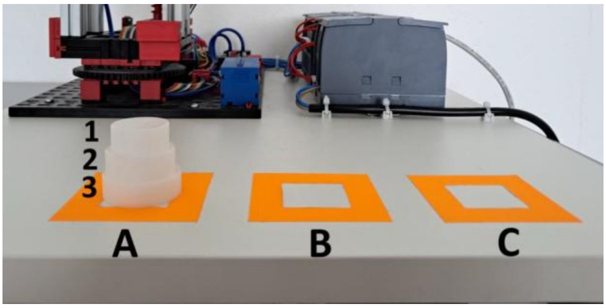
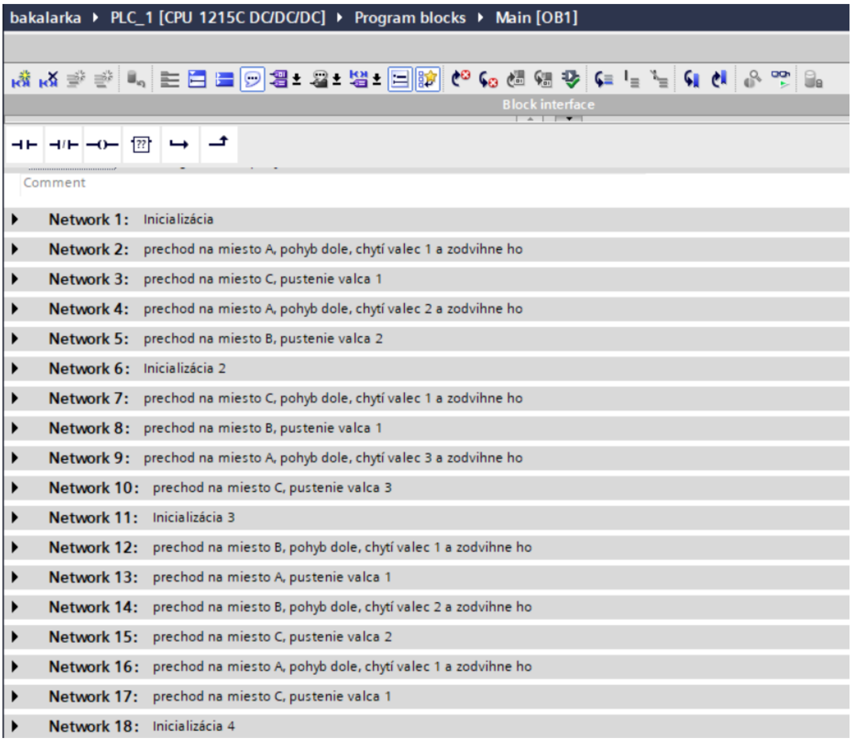


Fig.9. Position Markings and Cylinder Numbering.



bakalarka ▸ PLC_1 [CPU 1215C DC/DC/DC] ▸ Program blocks ▸ Main [OB1]

Network 1: Inicializácia
Network 2: prechod na miesto A, pohyb dole, chytí valec 1 a zodvihne ho
Network 3: prechod na miesto C, pustenie valca 1
Network 4: prechod na miesto A, pohyb dole, chytí valec 2 a zodvihne ho
Network 5: prechod na miesto B, pustenie valca 2
Network 6: Inicializácia 2
Network 7: prechod na miesto C, pohyb dole, chytí valec 1 a zodvihne ho
Network 8: prechod na miesto B, pustenie valca 1
Network 9: prechod na miesto A, pohyb dole, chytí valec 3 a zodvihne ho
Network 10: prechod na miesto C, pustenie valca 3
Network 11: Inicializácia 3
Network 12: prechod na miesto B, pohyb dole, chytí valec 1 a zodvihne ho
Network 13: prechod na miesto A, pustenie valca 1
Network 14: prechod na miesto B, pohyb dole, chytí valec 2 a zodvihne ho
Network 15: prechod na miesto C, pustenie valca 2
Network 16: prechod na miesto A, pohyb dole, chytí valec 1 a zodvihne ho
Network 17: prechod na miesto C, pustenie valca 1
Network 18: Inicializácia 4

Fig.10. A list of 18 names of the individual sections of the program (2: goto A and...,).

- **Initialization Sequence (Homing):** This is implemented as a separate function whose purpose is to move the manipulator to a precisely defined starting position. This position is detected by hardware using limit switches located at the ends of all three axes of motion. After all three sensors are successfully triggered, the system automatically activates the compressor required for the end-of-arm gripper to operate.
- **Continuous position calibration:** Since the manipulator is not equipped with absolute position sensors (measurement is performed only relatively using encoders), it was necessary to prevent the accumulation of positioning errors. For this reason, the initialization process is included in the control cycle a total of four times during a single complete program run.

- **Motion control and counters:** The remaining parts of the program control specific movements along individual axes as needed (the direction and target position are defined separately for each step, including suction cup control). The movement itself is controlled by evaluating pulses from the encoders:
    - A pulse signal from each encoder is fed directly to the input of the up counter (CTU – Counter Up).
    - Once the motor starts rotating in the desired direction, the counter registers the pulses. When the preset value is reached (the exact value of which was determined based on experimental measurements), the control system immediately shuts down the motor.
    - These counters are always reset during the initialization phase using a dedicated auxiliary variable.
- **Safety Stop:** A separate "Stop" function block has been created for emergency situations. When the "STOP" variable is activated (by writing a logical 1), this block immediately resets all output and key auxiliary variables to zero, thereby safely stopping program execution and equipment operation.

D. Data Processing for Visualization and Statistics

To maintain the clarity and simplicity of the machine's main control logic, all processes related to the operator panel have been separated into a separate functional block called "visualization." This block is called directly from the main program immediately after it starts.

Its primary function is to set the values of auxiliary variables, which are then assigned to HMI objects and dynamically control their visibility or state on the screen based on the current program phase. At the same time, the block processes operational statistics, which are stored in the user-defined data structure Data_vizualizacia:

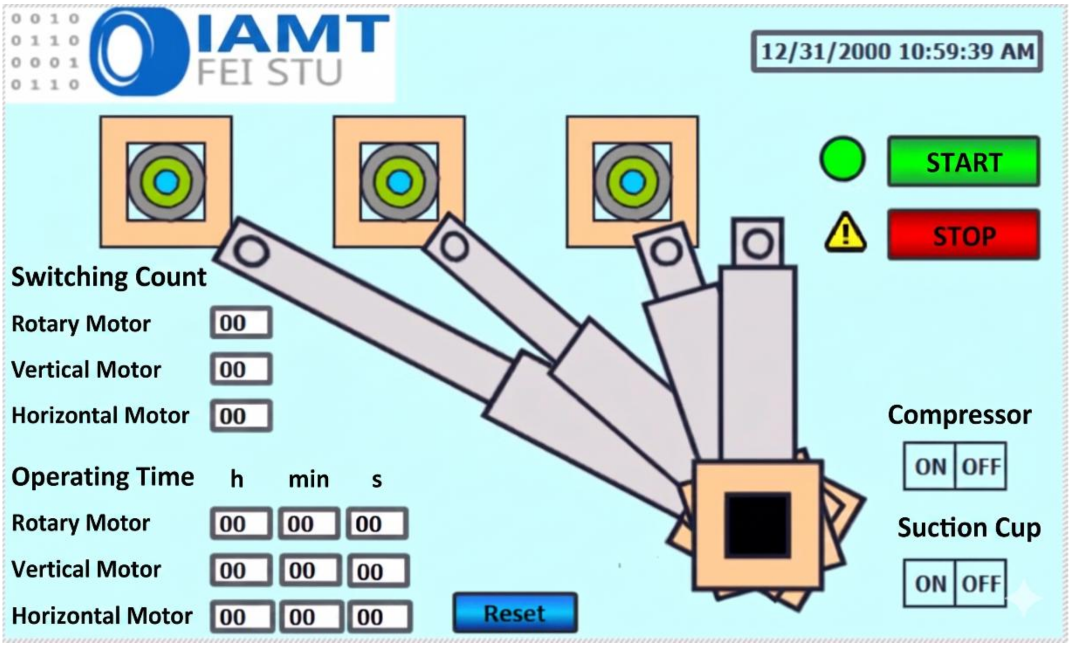
- **Number of motor starts:** The number of starts for each motor is tracked by detecting the rising edge of its output control signal. Each time the motor starts, the ADD mathematical instruction increments the corresponding variable in the data structure by 1.
- **Operating Hours (Run Time):** To measure the net run time of the motors, the PLC's system clock bit named Clock_1 Hz is used, which generates a stable pulse with a period of one second (this bit must first be enabled in the CPU's hardware configuration). By combining this clock signal, running motors, counters, and the ADD instruction, the system continuously records the exact operating time into a statistical structure.

E. Visualization Application

The graphical user interface (GUI) (Fig.11) was designed using WinCC Basic, which is a native component of the TIA Portal development environment. The KTP700 Basic PN operator panel was specified as the target hardware for the project.

To ensure communication with the control unit, the panel was assigned the static IP address 192.168.0.237 within the same subnet as the PLC. Subsequent debugging and verification of the visualization's functionality were performed using an integrated software simulator that fully emulates the physical panel directly on the development PC.

The visualization provides a clear 2D top-down view of the laboratory model of the manipulator. The interface is designed to provide the operator with immediate feedback on the status of the technology and intuitive control options.



12/31/2000 10:59:39 AM

Switching Count
Rotary Motor 00
Vertical Motor 00
Horizontal Motor 00

Operating Time h min s
Rotary Motor 00 00 00
Vertical Motor 00 00 00
Horizontal Motor 00 00 00

Fig.11. Display of the visualization screen.

Key elements of the graphical user interface:

- Object position monitoring: The cylinders being manipulated are graphically represented by circles with different diameters and colors. The system dynamically controls their visibility and displays their presence at three working positions in real time, based on the current program step.
- Manipulator animation: The manipulator’s movement is visualized using the principle of multiplexing (overlaying five different graphical states). Only the shape that exactly corresponds to the device’s current physical position is active, visible at any given time.
- Control and diagnostics: The "START" and "STOP" software buttons are used for basic cycle control and feature a visual indicator showing their active status.
    - In the lower right corner, there is a dynamic graphical display showing the status of the actuators (compressor and vacuum suction cup).
- Operational statistics: The bottom portion of the screen is reserved for displaying diagnostic data from the PLC. It shows the number of starts and the total runtime of each motor in the format hh:mm:ss. This data can be reset at any time using the dedicated button „Reset“.

5 Functional Testing and Verification

To comprehensively verify hardware integration, positioning accuracy, and the correctness of the control algorithm, a demonstration task known as the “Tower of Hanoi” was implemented. This process served to validate the interaction of all key components, particularly the encoders, the compressor, and the vacuum suction cup.

The layout of the components on the mounting board (PLC, power supply, and the manipulator itself) was spatially optimized to meet the requirements of this test task's working range.

A. Cycle Sequence and Control via the HMI

- **Initial state:** Since the system lacks camera vision or object presence sensors, the process requires the precise manual placement of three cylinders (in the order of largest, medium, smallest) at the center of the starting position „A".
- **Operation control:** The sequence is started by pressing the "START" button on the control panel. The process can be safely interrupted at any time by pressing the "STOP" button. To restart after an interruption, the rollers must be manually reset to their initial position.
- **Completion:** After successfully repositioning the entire tower to target position "C," the manipulator automatically returns to its home position.
- **Position display and diagnostics:** Due to the lack of real-time absolute position sensing, the HMI visualization displays the manipulator's step-by-step transitions between five defined reference positions. The panel also provides operational data (number of motor starts and runtime), which serves as the basis for predictive maintenance planning.

B. Technical Limitations and System Tuning

During implementation and testing, it was necessary to address several hardware limitations of the laboratory model through software:

- **No limit switches:** The device includes only switches for the home position. It does not include hardware limit switches for the working area, which placed high demands on the software collision protection.
- **Sensory efferent**: The vacuum suction cup is not equipped with a vacuum sensor, so the success of the grip depends entirely on the initial precise positioning of the rollers.
- **Path calibration:** Precise stopping of the manipulator at the desired points was achieved through iterative experimental tuning of the number of pulses and motor timing.

With optimal settings and uninterrupted operation, the control program can solve the Tower of Hanoi algorithm in the minimum theoretical number of 7 moves. The total time required to complete this fully automated cycle is 110 seconds..

Conclusion

The main objective of this thesis was to design and fully implement a control system for a laboratory model of a Fischertechnik three-axis manipulator. This objective was successfully achieved through the integration of a Siemens SIMATIC S7-1200 programmable logic controller and the development of an operator visualization application.

The practical implementation was divided into two main areas. The hardware component involved a detailed analysis of the manipulator's kinematics, the design of the component layout on the mounting board, and the precise mapping and interconnection of the PLC's input and output signals with the model's peripherals. This ensured reliable two-way communication. The software part was fully developed using the TIA Portal development environment. The control algorithm was developed in the standardized LD (Ladder Diagram) language, while the graphical interface for the KTP700 Basic HMI panel was designed in the WinCC Basic module and debugged using the integrated simulator.

To comprehensively verify the functionality and accuracy of the entire system, an algorithm for solving the Tower of Hanoi puzzle was implemented. The successful execution of this fully automated cycle in practice demonstrated that the chosen hardware architecture provides sufficient performance and flexibility for the smooth control and monitoring of the mechatronic system.

The workstation represents a highly versatile platform capable of performing a wide range of "Pick & Place" handling tasks using a vacuum gripper. The main benefit of this project lies in its educational potential. The fully configured and operational model serves as an excellent teaching aid for students, enabling them to learn how to work with SIMATIC systems and the TIA Portal environment. The ability to implement custom control programs and immediately debug them on real physical hardware provides invaluable feedback, thereby significantly enhancing the integration of theoretical knowledge with practical application in industrial automation.

Acknowledgement

This paper was supported by the Slovak Grant Agency VEGA 1/0637/23 and 1/0614/24 KEGA 021STU-4/2024, and by the Scientific Grant APVV-21-0125.

References

[1] M. Ceccarelli and E. Ottaviano, *Kinematic Design of Manipulators. Robot Manipulators*. 2008. ISBN 978-953-7619-06-0.

[2] C. Bernier, *Robotic Arms: Different Types and When to Use Them,* [Online]. Available: https://howto-robot.com/expert-insight/robotic-arms. [Accessed: 2025-05-29].

[3] *Robotické manipulátory – priemyselné roboty,* [Online]. Available: https://strojsnv.edupage.org/files/3-Roboticke_manipulatory_-_priemyselne_roboty.pdf. [Accessed: 2025-05-29].

[4] *How to Select the Right Motor for Your Robot*, [Online]. Available: https://www.roboticstomorrow.com/article/2021/06/how-to-select-the-right-motor-for-your-robot/17070. [Accessed: 2025-05-29].

[5] *What Types Of Motors Are Used in Industrial Robotics?,* [Online]. Available: https://www.sitmotors.com/news/what-types-of-motors-are-used-in-industrial-ro-73797511.html. [Accessed: 2025-05-29].

[6] Ľ. Dorčák, J. Terpák, and F. Dorčáková, *Teória automatického riadenia: Spojité lineárne systémy*. 2006. ISBN 80-8073-025-3.

[7] J. Cigánek and F. Žemla, *PLC systémy v mechatronike*. 2023. ISBN 978-80-8208-097-4.

[8] P. Kuna, *Základy automatizácie*. 2015. ISBN 978-80-558-0841-3.

[9] *Essential Components of PLCs Explained,* [Online]. Available: https://kwoco-plc.com/plc-components/. [Accessed: 2025-05-29].

[10] *Didactic booklet - Vakuum Sauggreifer 24V*, [Online]. Available: https://www.fischertechnik.de/-/media/fischertechnik/fite/service/elearning/simulieren/doku-24v/de/536630-vakuum_sauggreifer_24v.pdf. [Accessed: 2025-05-29].

[11] Y. Aakash, *What is Arduino? A Beginner's Guide to Understand it,* [Online]. Available: https://hackr.io/blog/what-is-arduino. [Accessed: 2025-05-29].

[12] *Raspberry Pi Introduction*, [Online]. Available: https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction. [Accessed: 2025-05-29].

[13] *Introduction to Programmable Logic Controllers: Your Gateway to Industrial Automation,* [Online]. Available: https://kwoco-plc.com/plc-guide/. [Accessed: 2025-05-29].

[14] *Top PLC Manufacturers: The Best Brands in Programmable Logic Controllers*, [Online]. Available: https://kwoco-plc.com/plc-manufacturers/. [Accessed: 2025-05-29].

[15] *SCADA*, [Online]. Available: https://www.paessler.com/it-explained/scada. [Accessed: 2025-05-29].

[16] *What are HMI and SCADA systems and how are they used in industrial automation?,* [Online]. Available: https://vakoms.com/blog/what-are-hmi-and-scada-systems-and-how-are-they-used-in-industrial-automation/. [Accessed: 2025-05-29].

[17] *Vakuum-Sauggreifer 24V - Fischertechnik*, [Online]. Available: https://www.fischertechnik.de/de-de/produkte/industrie-und-hochschulen/simulationsmodelle/536630-vakuum-sauggreifer-24v. [Accessed: 2025-05-29].

[18] *Technisches Datenblatt Encoder Motor 24V,* [Online]. Available: https://fiproductmedia.azureedge.net/media/Certification%20Documents/Technical%20Datasheets/fischertechnik/TDB_144643-ENCODERMOTOR24V.pdf. [Accessed: 2025-05-29].

Authors

Ing. Marián Choma
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
marian.choma@stuba.sk

He received his Master’s degree in Applied Informatics from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, in 2024. Currently, he is pursuing his PhD at the Institute of Automotive Mechatronics (FEI STU). His research primarily focuses on artificial intelligence—specifically computer vision and camera systems—as well as software architecture.

Bc. Ján Ulej
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ulej@stuba.sk

He received his Bachelor’s degree in Automotive Mechatronics from the Faculty of Electrical Engineering and Information Technology at the Slovak University of Technology (FEI STU) in Bratislava in 2025. He is currently pursuing a Master’s degree in Applied Mechatronics and Electromobility, with a focus on intelligent technologies and systems.

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
jan.ciganek@stuba.sk

He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.

⇨

Bc. Hortenzia Wollentová
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
xwollentova@stuba.sk

She received her bachelor’s degree from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, and she is currently pursuing an engineering degree in Applied Mechatronics and Electromobility. Her professional interests focus on SCADA systems.