ISSN (e): 2250 – 3005 || Volume, 15 || Issue, 6|| June – 2025 ||
International Journal of Computational Engineering Research (IJCER)

Different AI Tools for PLC Programming in Industrial Engineering Education

Gordana Ostojic[1], Stevan Stankovski[2]
[1]Full professor,
Faculty of Technical Sciences,
University of Novi Sad, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia.
[2] Full professor,
Faculty of Technical Sciences,
University of Novi Sad, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia.
Corresponding Author: Gordana Ostojic

ABSTRACT
A industrial engineering education is crucial for modern industry and society. Industrial engineers are well placed to bridge the gap between engineering, management and technology. They optimise processes, improve productivity and ensure the efficient use of resources. They play a vital role in designing, analysing and improving systems involving people, materials, information, equipment and energy. In an era of increasing global competition, rapid technological change, and the imperative of sustainable operations, the role of industrial engineers is becoming ever more significant. Through proper education, these professionals gain the necessary knowledge and tools to solve complex problems, reduce costs, enhance product quality and create safer, more ergonomic work environments. In smart production systems, where automation is highly advanced, one of the most important skills for industrial engineers is PLC programming. This paper analyses a typical control task in industrial engineering education in relation to modern approaches to PLC programming using AI tools such as ChatGPT, Copilot and Writesonic.
KEYWORDS: *Education, PLC, programming, AI tools, control tasks*

Date of Submission: 15-06-2025 Date of acceptance: 30-06-2025

I. INTRODUCTION

Industrial engineering is a multidisciplinary field that focuses on the optimization of complex systems, processes, and organizations. It combines engineering principles with management practices to improve productivity, efficiency, and quality in production and service systems.

Industrial engineers are involved in designing workflows, optimizing supply chains, improving manufacturing processes, and integrating new technologies into production systems. Their work ensures that resources such as materials, energy, machines, and labor are used in the most efficient way possible [1].

Through the use of methods like process analysis, simulation, lean manufacturing, and data-driven decision-making, industrial engineers play a key role in increasing competitiveness and sustainability in modern industries. The field is essential for companies aiming to reduce costs, improve quality, and meet the growing demands of global markets.

One of the essential skills for industrial engineers, especially in modern automated production systems, is PLC (Programmable Logic Controller) programming. PLCs represent the core of automation in manufacturing facilities, controlling machines, production lines, and entire industrial processes [2]. Understanding PLC programming enables industrial engineers to actively participate in the design, implementation, and optimization of automated systems.

By mastering PLC programming, industrial engineers can: communicate effectively with automation engineers and technicians, optimize production processes through automation, reduce downtime and improve process reliability, and support the implementation of Industry 4.0 concepts in factories.

Integrating PLC programming into the education of industrial engineers strengthens their practical skills and prepares them for the technological challenges of modern industry. This combination of systems thinking, process optimization, and automation knowledge makes industrial engineers highly valuable in today's competitive industrial environment.

II. PLC PROGRAMMING EDUCATION

PLC programming is one of the key courses taught in the Industrial Engineering study program at the Faculty of Technical Sciences, University of Novi Sad. Students are trained to work with various types of equipment and different programming approaches, with special emphasis placed on programming in Structured Text (ST).

The primary reason for focusing on Structured Text is its versatility and similarity to high-level programming languages, which makes it particularly suitable for solving complex automation tasks. Structured Text allows for writing clear, efficient, and scalable code, especially when dealing with advanced logical operations, mathematical calculations, loops, and decision-making algorithms.

By learning Structured Text, students not only acquire the skills necessary for modern industrial automation but also develop programming habits that are transferable to other areas of engineering and software development. This approach ensures that graduates are well-prepared to meet the demands of Industry 4.0, where flexibility, precision, and integration of automated systems are crucial for success.

With the rapid development of artificial intelligence, tools such as ChatGPT [3], Microsoft Copilot [4], and Writesonic [5] have become valuable resources in the field of education, including technical and engineering disciplines [6]. These AI-powered platforms assist users in generating text, writing code, and providing intelligent suggestions during problem-solving. They can also be used for PLC programming, because these tools can significantly enhance the learning experience for students of industrial engineering. They provide instant support in several ways: generating example code for specific automation tasks (e.g., sequences, error handling, etc.), explaining complex programming concepts in a clear and understandable manner, assisting in debugging by offering suggestions on how to correct syntax errors or improve logic.

By using tools like ChatGPT, students can ask for explanations or practical examples of PLC logic [6], while Copilot can integrate with coding environments to assist directly in writing PLC code where supported. Writesonic can help with generating reports, documentation, or structured descriptions of control processes.

The integration of AI tools into engineering education promotes self-directed learning, encourages experimentation, and bridges the gap between theoretical knowledge and practical application. For future industrial engineers, familiarity with both automation systems and AI-powered development assistance will be an essential advantage in modern smart manufacturing environments.

For all the reasons mentioned above, during the exercises in PLC programming course, the students explored writing code for the same task using the traditional method described in the textbook [7], as well as by utilizing the three AI tools mentioned. PLC programming training starts from simple tasks to complex ones.

One of the tasks that was given to students to solve in traditional manner and with the usage of AI tools is the following: "On one press machine, three workpieces are simultaneously cut from a sheet metal piece. This is achieved by operating three cutting cylinders in parallel. In order to achieve optimal quality of the cut parts, it is necessary to ensure that the three cylinders operate as simultaneously as possible. To enable visual monitoring of their synchronization, appropriate indicators have been installed.

Three indicator lamps (H1, H2, H3) are used to show how many cutting tools have reached their bottom end position. The signaling is performed using limit switches B1, B2, and B3. Lamp H1 should light up when only one of the tools has reached the bottom end position, H2 should light up only when two tools have reached the bottom end position, and H3 should light up when all three tools have reached the bottom end position.

All three cylinders are activated by the same button (START button). After releasing the START button, the cylinders return to their initial positions. The speed of the piston rod retraction for each cylinder can be adjusted using a throttle check valve."

The traditional solution that is presented in the textbook [7] written regarding IEC 61131-3 standard, is next:

```
PROGRAM ZadatakX
VAR_INPUT
  Start AT %IX0.0 : BOOL; //START button for cylinder control
  B1 AT %IX0.1 : BOOL; //B1: tool 1 in reached end position
  B2 AT %IX0.2 : BOOL; //B2: tool 2 in reached end position
  B3 AT %IX0.3 : BOOL; //B3: tool 3 in reached end position
END_VAR
VAR_OUTPUT
  Y1 AT %QX0.0 : BOOL := 0; //Control signal for cylinder 1
  Y2 AT %QX0.1 : BOOL := 0; //Control signal for cylinder 2
  Y3 AT %QX0.2 : BOOL := 0; //Control signal for cylinder 3
  H1 AT %QX0.3 : BOOL := 0; //H1: Lamp - 1 cylinder has reached end position
  H2 AT %QX0.4 : BOOL := 0; //H2: Lamp - 2 cylinder has reached end position.
  H3 AT %QX0.5 : BOOL := 0; //H3: Lamp - 3 cylinder has reached end position.
```

```
END_VAR

Y1 := Start;
Y2 := Start;
Y3 := Start;
H1 := (NOT B1 AND NOT B2 AND B3) OR (NOT B1 AND B2 AND NOT B3) OR (B1 AND NOT B2 AND
NOT B3);
H2 := (NOT B1 AND B2 AND B3) OR (B1 AND NOT B2 AND B3) OR (B1 AND B2 AND NOT B3);
H3 := B1 AND B2 AND B3;
```

The solution suggested by ChatGPT is as follows:

```
PROGRAM PressControl
VAR
    START     : BOOL;     // Input: START button (active when pressed)
    B1        : BOOL;     // Input: Limit switch for cylinder 1 (TRUE = reached end position)
    B2        : BOOL;     // Input: Limit switch for cylinder 2 (TRUE = reached end position)
    B3        : BOOL;     // Input: Limit switch for cylinder 3 (TRUE = reached end position)

    CYL1      : BOOL;     // Output: Control signal for cylinder 1
    CYL2      : BOOL;     // Output: Control signal for cylinder 2
    CYL3      : BOOL;     // Output: Control signal for cylinder 3

    H1        : BOOL;     // Output: Lamp - 1 cylinder has reached end position
    H2        : BOOL;     // Output: Lamp - 2 cylinders have reached end position
    H3        : BOOL;     // Output: Lamp - all 3 cylinders have reached end position

    Count     : INT;      // Auxiliary variable for counting how many cylinders have reached the end
END_VAR

// Activate cylinders when START button is pressed
IF START THEN
    CYL1 := TRUE;
    CYL2 := TRUE;
    CYL3 := TRUE;
ELSE
    // When START button is released, cylinders return to starting position
    CYL1 := FALSE;
    CYL2 := FALSE;
    CYL3 := FALSE;
END_IF

// Count how many cylinders have reached their end position
Count := 0;
IF B1 THEN
    Count := Count + 1;
END_IF;
IF B2 THEN
    Count := Count + 1;
END_IF;
IF B3 THEN
    Count := Count + 1;
END_IF;

// Control the signal lamps based on the number of cylinders that have reached the end
H1 := (Count = 1);    // H1 ON → 1 cylinder reached
H2 := (Count = 2);    // H2 ON → 2 cylinders reached
H3 := (Count = 3);    // H3 ON → all 3 cylinders reached

END_PROGRAM
```

The solution suggested by Copilot is as follows:

```
PROGRAM PressControl

VAR
   START      : BOOL;     // Button to start the cycle
   LS1, LS2, LS3 : BOOL;   // Limit switches for cylinder end positions
   Lamp1, Lamp2, Lamp3 : BOOL; // Indicator lamps
   Cylinder1, Cylinder2, Cylinder3 : BOOL; // Outputs to activate cylinders
   ToolCounter : INT;     // Counter for tools in lower end position
END_VAR

// Activate cylinders when START is pressed
IF START THEN
   Cylinder1 := TRUE;
   Cylinder2 := TRUE;
   Cylinder3 := TRUE;
ELSE
   Cylinder1 := FALSE;
   Cylinder2 := FALSE;
   Cylinder3 := FALSE;
END_IF;

// Count how many cylinders have reached end position
ToolCounter := INT_TO_DINT(LS1) + INT_TO_DINT(LS2) + INT_TO_DINT(LS3);

// Light up lamps based on number of tools in position
CASE ToolCounter OF
   1:
      Lamp1 := TRUE;
      Lamp2 := FALSE;
      Lamp3 := FALSE;
   2:
      Lamp1 := FALSE;
      Lamp2 := TRUE;
      Lamp3 := FALSE;
   3:
      Lamp1 := FALSE;
      Lamp2 := FALSE;
      Lamp3 := TRUE;
ELSE
      Lamp1 := FALSE;
      Lamp2 := FALSE;
      Lamp3 := FALSE;
END_CASE;
```

The solution suggested by Writesonic Chatsonic is as follows:

```
PROGRAM CuttingProcess
VAR
   Tool1Position : BOOL;  // Position of Tool 1
   Tool2Position : BOOL;  // Position of Tool 2
   Tool3Position : BOOL;  // Position of Tool 3
   StartButton : BOOL;    // Start button state
   IndicatorH1 : BOOL;    // Indicator for 1 tool in position
   IndicatorH2 : BOOL;    // Indicator for 2 tools in position
   IndicatorH3 : BOOL;    // Indicator for all 3 tools in position
END_VAR

// Main Program Logic
```

```
IF StartButton THEN
    // Activate all cylinders
    Tool1Position := FALSE; // Assume not in position
    Tool2Position := FALSE; // Assume not in position
    Tool3Position := FALSE; // Assume not in position

    // Simulate tool movement and check positions
    IF (Tool1Position = TRUE) AND (Tool2Position = FALSE) AND (Tool3Position = FALSE) THEN
        IndicatorH1 := TRUE;
        IndicatorH2 := FALSE;
        IndicatorH3 := FALSE;
    ELSIF (Tool1Position = TRUE) AND (Tool2Position = TRUE) AND (Tool3Position = FALSE) THEN
        IndicatorH1 := FALSE;
        IndicatorH2 := TRUE;
        IndicatorH3 := FALSE;
    ELSIF (Tool1Position = TRUE) AND (Tool2Position = TRUE) AND (Tool3Position = TRUE) THEN
        IndicatorH1 := FALSE;
        IndicatorH2 := FALSE;
        IndicatorH3 := TRUE;
    END_IF;
END_IF;

// Reset Logic
IF NOT StartButton THEN
    // Reset all positions and indicators
    Tool1Position := FALSE;
    Tool2Position := FALSE;
    Tool3Position := FALSE;
    IndicatorH1 := FALSE;
    IndicatorH2 := FALSE;
    IndicatorH3 := FALSE;
END_IF;
```

III. DISCUSION

The traditional solution presented in the textbook follows the IEC 61131-3 standard and relies on direct Boolean logic to control the output indicators. This method explicitly defines all possible combinations of the three limit switches (B1, B2, B3) to determine whether one, two, or all three cylinders have reached their end positions. While this approach is fully compliant with industrial standards and offers deterministic control suitable for PLC environments, it becomes increasingly difficult to maintain as system complexity grows. The Boolean expressions, although precise, are harder for students to read and understand, particularly when multiple conditions are involved. This method is valuable for learning the basics of PLC programming and developing an understanding of structured programming logic in automation systems.

The solution provided by ChatGPT introduces an improved structure by using an integer counter to track how many cylinders have reached the end position. Instead of writing complex Boolean expressions for each possible combination, the program simply increments the counter for each activated limit switch and assigns output indicators based on the counter value. This significantly improves readability and makes the program easier to expand if more sensors or conditions need to be added later. For students, this method is intuitive, aligns with general programming practices, and promotes modular thinking and abstraction, which are essential skills in modern industrial automation. Although slightly more abstract than direct Boolean logic, it represents a step toward scalable and maintainable code development.

The solution generated by Copilot builds on the same concept of using a counter but introduces a CASE structure to control the output signals. This approach is very readable and structured, making it easy to understand and modify. The CASE structure clearly associates each possible number of active tools with its corresponding lamp output, making the code systematic and neatly organized. However, the inclusion of unnecessary data type conversions, such as the conversion of Boolean values to DINT, adds a slight complexity that could be avoided in simpler PLC programs. Despite that, the structured format and clarity make this solution very appropriate for both industrial practice and educational use, particularly for teaching good programming habits in automation.

On the other hand, the solution proposed by Writesonic takes a different approach, more aligned with simulation or conceptual modeling. The code assumes the tool positions are false at the beginning of each cycle and simulates their states within the same block, which is not practical for real-time PLC control. Although the logic of indicator activation is similar in structure to the other solutions, the handling of inputs and outputs is oversimplified and would require significant corrections for practical implementation on real industrial hardware. While this approach may help students conceptually understand process behavior, it is not recommended as an example for learning proper PLC programming.

IV. CONCLUSION

Considering the analysis of the presented solutions, it is clear that combining traditional and modern approaches to PLC programming provides significant educational value for students of industrial engineering. The traditional method, based on the IEC 61131-3 standard and presented in the textbook, offers students a solid foundation in structured automation programming. It familiarizes them with the formal logic expected in industrial environments and strengthens their understanding of how PLCs handle discrete control tasks.

However, the solutions generated by advanced AI tools such as ChatGPT and Copilot demonstrate the benefits of modern programming practices, particularly in terms of readability, scalability, and adaptability. These solutions promote the development of algorithmic thinking, modular programming, and the use of structures like counters and CASE statements, which are essential skills in today’s increasingly complex and digitalized industrial systems.

By exposing students to both traditional programming techniques and AI-assisted coding tools, they not only gain technical competence but also develop flexibility in problem-solving, critical thinking, and adaptation to new technologies. Additionally, the integration of AI tools into the learning process aligns with the principles of Industry 4.0 and prepares future engineers for the challenges of modern smart manufacturing.

Therefore, using a combination of textbook-based methods and AI-assisted approaches in PLC programming education ensures that students are equipped with both the theoretical knowledge and practical skills needed for success in the evolving field of industrial automation.

ACKNOWLEDGMENT

The research for this paper was funded by the project “Unapređenje kvaliteta nastave na studijskim programima Departmana kroz implementaciju rezultata naučno-istraživačkog rada u oblasti Industrijskog inženjerstva i menadžmenta.”

REFERENCES

[1]. M. Lazarevic, Dj.Vukelic, S. Stankovski, N. Sremčev and G. Ostojić G, Enhancing Student Competence through Novel Student Enterprise Concept, *International Journal of Engineering Education*, Vol. 33, (2017), No. 6(A), pp. 1824-1833.

[2]. N. T. Cuong, Using PLC to Control an Automatic Flat Fabric Pressing Machine, International Journal of Computational Engineering Research (IJCER) Vol. 14, (2024), No. 6, pp. 106-111.

[3]. https://chatgpt.com/, Accessed: 2.6.2025.

[4]. https://copilot.microsoft.com/, Accessed: 3.6.2025.

[5]. https://writesonic.com/, Accessed: 2.6.2025.

[6]. S. Stankovski, G. Ostojic and G. Sinikovic, Transformation of Learning in Higher Education based on AI-tools, *Journal of Mechatronics, Automation and Identification Technology – JMAIT* Vol. 9, (2024), No. 3, pp. 17-20

[7]. S. Stankovski, S. Tegeltija, M. Stanojevic, M. Babic, G. Ostojic, M. Rakovic, L. Tarjan, I. Senk and M. Nikolic, Zbirka resenih zadatka iz: Programiranje i primena programabilnih logičkih kontrolera, Fakultet tehničkih nauka u Novom Sadu, Novi Sad, 2023.