FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 17, N^o 3, 2019, pp. 405 - 414
https://doi.org/10.22190/FUME190123033B

Original scientific paper

AUTOMATIC GENERATION OF THE PLC PROGRAMS FOR THE SEQUENTIAL CONTROL OF PNEUMATIC ACTUATORS

Vladislav Blagojević[1], Saša Ranđelović[1],
Vlastimir Nikolić[1], Slobodan Dudić[2]

[1]Faculty of Mechanical Engineering, University of Niš, Niš, Serbia
[2]Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia

Abstract. *Nowadays, programmable logic controllers (PLC) are widely used in many automated systems, especially for the control of various actuators. The most common PLC programming is performed by either using a ladder diagram or a structured text. The paper presents the automatic generation of PLC programs for the purpose of sequentially controlling pneumatic actuators. In this paper, the pneumatic actuators are supplied and controlled by 5/2-way as well as 5/3-way bistable pneumatic valves with electric activation. The valve type depends on the number of positions in which the actuator should come, and the position sensors are used for detecting its movement. The characteristic encoding of the movement of actuators, position sensors and control commands is performed. The advantages of the automatic generation of the PLC commands and the entire program described in this paper are illustrated in a real example.*

Key Words: *PLC, Programming, Sequential Control, Actuator*

1. INTRODUCTION

Wherever it is necessary to automate various processes, programmable logic controllers (PLC) are used. They are developed to provide flexibility and replace the old ways of controlling. There are various manufacturers of PLCs, such as SIEMENS, OMRON, FESTO, MITSUBISHI, ABB, etc. Also, there are many different PLC types, from small devices in housings integrated with the processor, to large rack-mounted modular devices, with numerous inputs and outputs [1,2]. Nowadays, PLCs are programmed by adequate PC software. This software enables programming by using a ladder diagram and a structured text.

Some authors have dealt with the issue of automatic PLC programs generation using adequate PC software as well as sequential control systems. Dworzak and Mikulczyński

Received January 23, 2019 / Accepted July 25, 2019

Corresponding author: Vladislav Blagojević
University of Niš, Faculty of Mechanical Engineering, A. Medvedeva 14, 18000 Niš, Serbia
E-mail: vlada@masfak.ni.ac.rs

[3] introduced a new method for the synthesis of sequential control algorithms for pneumatic actuators controlled by monostable valves. Malayappan el al. [4] used a cascading method in designing a sequential control system for an industrial robot. Mroz and Brol [5] developed a sequential control strategy for a pneumatic-hydraulic actuator, as well as its simulation model. They even increased the energy efficiency of the proposed system. Bayoumi [6] presented a new tree-chart method to design a sequential logic controller with or without an intermediate stop of an actuator. Hasdemir and Kurtulan [7] and Salunke et al. [8] introduced a methodology for expressing an automaton in logical domain in the Matlab program that automatically generates PLC commands. Guttel et al. [9] focused on the safety functions, start-up and shutdown sequences of the control code and strived for a concept to automatically generate that part of the control. Schumacher and Fay [10] and Schumacher et al. [11] used GRAFCET as a modeling language for automatic generation of the control code. Julius et al. [12] made a systematic approach to automatically transform GRAFCET into the PLC code while retaining the hierarchical structures. Wcislik et al. [13,14] presented a new algorithm for the programming of sequential control systems in the LabVIEW software. They implemented the language of function block diagrams in PLCs. Adiego et al. [15] introduced a general automated methodology for formal verification of PLC programs. The methodology is based on an intermediate model, used as a pivot between all the PLC and formal modeling languages. Qamsane et al. [16] presented a model-driven engineering method, which aims at automatically generating distributed PLC-Based control of automated manufacturing systems in the form of GRAFCET. Blagojevic et al. [17] presented an algorithm for the PLC programming of the sequential control systems of pneumatic actuators with a single branch and only two end positions.

This paper presents a new way of automated PLC programming of sequential pneumatic asynchronous automata control systems with parallel and single branches. This way of the PLC programming enables a unique designation of pneumatic actuators and their movement according to certain established technological processes in steps, in order of their execution. The advantage of the algorithm is a possibility to control the stopping of actuators not only in the final but also in the middle positions. Furthermore, it is possible to automatically generate PLC commands and the entire program for a sequential pneumatic control system with parallel and single branches. Microsoft Office Excel was used for the implementation of the automatic generation of PLC commands.

2. Pneumatic Actuator System

Usually a pneumatic actuator system consists of a pneumatic actuator and a 5/2-way bistable direction valve, for the simple actuator movement only between end positions, Fig. 1, and a 5/3-way bistable direction valve, for the actuator movement and stopping in the middle positions, Fig. 2. Therefore, two command signals are required for each actuator, y_i+ and y_i-, for the movement of the actuator in one and the opposite direction, where i=1, 2, ..., n is the actuator number.

Every actuator has two end position sensors and in some cases, more middle position sensors. Position sensors are used to detect the movement and position of the actuator. For pneumatic actuator C_i with two end positions, signal x_{i0} represents the initial position and x_{i1} the final position, Fig. 1. If actuator C_i has to stop somewhere in the middle

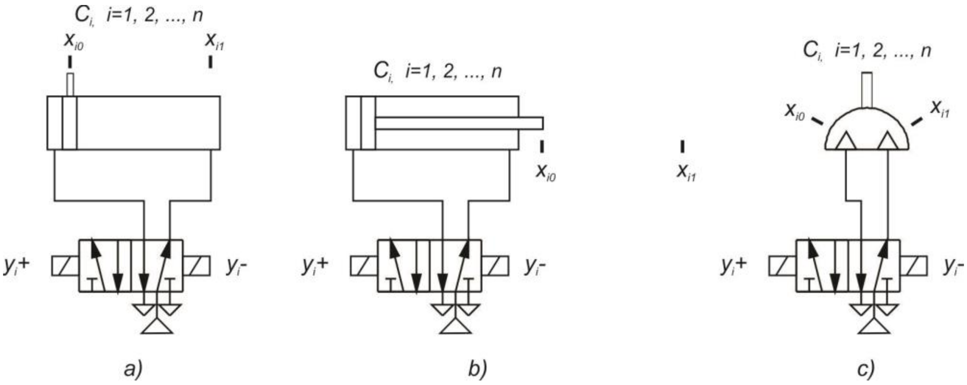


Fig. 1 Pneumatic system with two end positions with: a) Symmetric actuator, b) Asymmetric actuator, c) Semi-rotary drive

positions of its stroke, there is a need for *p* position sensors, $p>1$. The signal from position sensor x_{i0} is for the initial position, x_{i1}, x_{i2}, ... $x_{i(p-1)}$ are for the middle positions and x_{ip} is for the final position, Fig. 2.

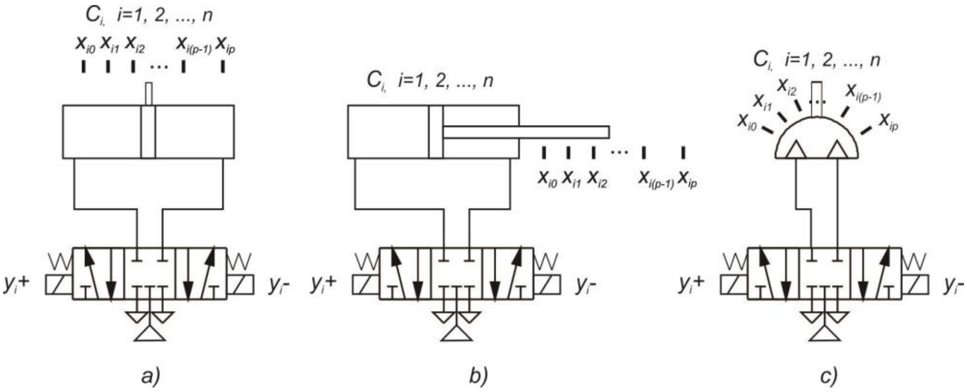


Fig. 2 Pneumatic system with more than two end positions with: a) Symmetric actuator, b) Asymmetric actuator, c) Semi-rotary drive

3. Sequential Control Pneumatic System

A sequential control system is a control system with individual steps, which are processed in a predetermined order, where the progression from one step to the next depends on the matching defined conditions. If pneumatic actuators are used to perform the operations, these systems are called sequential control pneumatic systems. The pattern of a sequential control system can have only a single branch, Fig. 3a, parallel branches, Fig. 3b, or a combination of the previous two.

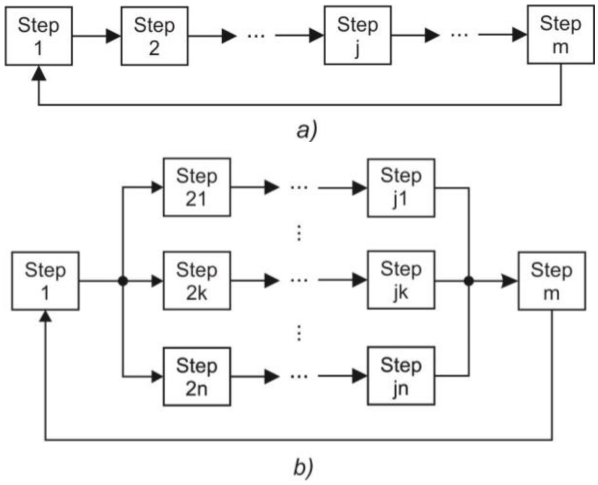


Fig. 3 Sequential control system with: a) single branch, b) parallel branches

In a pneumatic sequential control system with a single branch and with several different pneumatic actuators, every step consists of one movement of the actuator [17]. In a sequential control system with parallel branches it is possible to simultaneously execute more steps.

In pneumatic sequential control systems, the designation of the movement of actuators between only two end positions is:

- C_i+, for the extend direction of movement for linear actuators and the right direction of movement for semi-rotary drives, and,
- C_i-, for the retract direction of movement for linear actuators and the left direction of movement for semi-rotary drives.

If actuator C_i has to stop somewhere in a middle position of its stroke, the designation of its movement is:

- C_i+j, for the extend direction of movement for linear actuators and the right direction of movement for semi-rotary drives, and,
- C_i-j, for the retract direction of movement for linear actuators and the left direction of movement for semi-rotary drives,

where j=0, 1, …, p is the number of the position sensor that has to be reached by the actuator. The minimum number p=1 for only two end position sensors of the actuator, and p>1 if there are some position sensors in the middle of the actuator stroke.

This paper considers the case of sequential control of the pneumatic actuators with two end positions as well as several middle positions. The pattern of the actuator movement is a combination of systems with a single and several parallel branches.

4. Automatic Generation of a PLC program

Automatic PLC programming of the sequential control of pneumatic actuators consists of three steps, Fig. 4. The first step is to describe the actuator movement, the second step is to generate input and output addresses for sensor and command signals, and the final step is to generate PLC commands and the entire program.

For the automatic generation of PLC commands and the entire program it is necessary to perform the exact description of the actuator movement according to sequential control, step by step. The best way is by using tables, where the number of sequential control branches is equal to the number of columns and the number of sequential control steps is equal to the number of rows, Fig. 5. The movement of each actuator is entered into one cell of the table. Also, it is very important to describe the actuator movement, as it was explained in Chapter 3, with C_i+j and C_i-j, where the first two characters designate the number of actuators, the "+" or "-" sign stands for the direction of the actuator movement and the last symbol representing the number of the position sensors that have to be reached at the end of the actuator movement. It is important that every actuator must return to the start position at the end of the whole sequential control cycle.

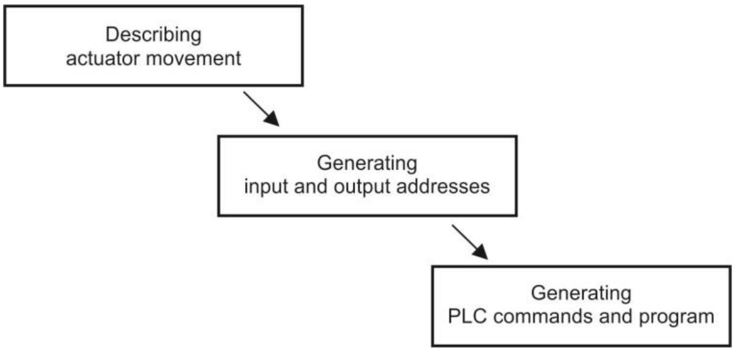
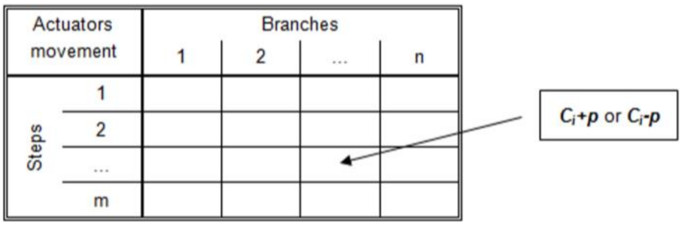


Fig. 4 Block scheme of automatic PLC programming of the sequential control of pneumatic actuators

This way of describing the movement of actuators enables an easy generation of the output command signals, from the first three characters, and the input signals from the last character.



Actuators movement		Branches			
		1	2	...	n
Steps	1				
	2				
	...				
	m				

C_i+p or C_i-p

Fig. 5 Table for describing actuator movement

After the first step of automatic PLC programming of the sequential control of pneumatic actuators, based on the given movement of the actuators and the analysis of every single character of description, in the second step it is possible to generate the necessary input and output addresses for the sensor and command signals. In this paper, the designation of input signals is Ix.x and output signals is Ox.x, from the FESTO standard for PLC programming. The character or characters between character "C" and signs "+" or "-", in the description of the actuator movement from the first step, stands for the number of actuators. Now it is very easy to generate a list of output addresses for command signals for each actuator, for example: if “C_i+_p” is written then the output signal is “y_i+”, or if “C_i-_p” is written then the output signal is “y_i-”, where i is the actuator number and p is the sensor number. For example, the list of output addresses is:

O0.1 y_1+	O1.1 y_1-
O0.2 y_2+	O1.2 y_2-
…	
O0.n y_n+	O1.n y_n-

The characters after signs “+” or “-“, in the description of the actuator movement, represent the number of the position sensors that have to be reached at the end of the actuator movement. After analyzing the number of actuators and the number of position sensors, it is possible to generate a list of input addresses from sensors for each actuator, for example: if “C_i+_p” or “C_i-_p” is written then the input signal is “x_{ip}”, where i is the actuator number and p is the sensor number. For example, the list of input addresses is:

I0.1 x_{10}	I1.1 x_{20}	…	In.1 x_{n0}
I0.2 x_{11}	I1.2 x_{21}	…	In.2 x_{n1}
	…		
I0.p x_{1p}	I1.p x_{2p}	…	In.p x_{np}

The final step in automatic PLC programming of sequential control is generation of PLC commands and the entire program. Analyzing the description of the movement of actuators from the table of the first step, Fig. 5, it is possible to detect the conditions for executing each step of sequential control. The condition for the execution of the next step of sequential control is that the previous step is finished. The verification of the completion of the previous step consists of detecting the existence of the signal of the sensor that has to be reached in the previous step. For example, if there are two steps, STEP 1: C_1+_1 and STEP 2: C_1-_0, then the condition for the execution of STEP 2 is the existence of the signal of sensor x_{11}, and the execution command for STEP 2 is y_1-. In a sequential control system with a single and parallel branches the condition for the execution of the next step, if there are parallel branches before it, is a logical product of all signals of the sensors that are reached in the previous step. For example, if there are two steps, where STEP 1 consists of three branches, STEP 1: C_1+_1 C_2+_2 C_3+1, and STEP 2: C_2-_0, then the condition for the execution of STEP 2 is the existence of all the signals of sensors x_{11}, x_{22} and x_{31}, and the execution command for STEP 2 is y_2-.

After detecting all conditions from the sequential control cycle, it is possible to generate a PLC program. The best way for generating a PLC program is in the form of structured text by step operation, supported by the IEC 61131-3 standard [2,18].

For this purpose, the syntax of the structured text of the PLC program is:

```
STEP K
IF
    Signal from the position sensor of the previous actuator movement exists
THEN
    SET command for equal actuator movement in the right direction
    RESET command for equal actuator movement in the opposite direction
```

where **K** is the number of the step.

In the first STEP of the PLC program there is always a minimum of two conditions. The first condition is **START** or **START CONDITIONS** and the other condition depends on the sequential control of actuators.

At the end of the PLC program, after the final STEP operation, the JMP TO STEP 1 command is generated. This command enables the program to return to the start of the first STEP.

5. AUTOMATIC GENERATION OF A PLC PROGRAM USING MICROSOFT OFFICE EXCEL

In this paper, Microsoft Office Excel is used for the automatic generation of PLC commands. The main reason for using Excel is that sequential control steps are very easy to describe by a spreadsheet, which is adjustable for text and numeric processing as well. Also, there are many commands that are useful in processing the description of the movement of actuators, recognizing the number of actuators, direction of movement and number of position sensors. Finally, Microsoft Office Excel can be applied on Windows, macOS, Android and iOS platforms, and it is available on various devices, such as PCs, Tablets, smartphones, etc.

Automatic PLC programming of the sequential control of pneumatic actuators by Excel consists of three parts, Fig. 6.

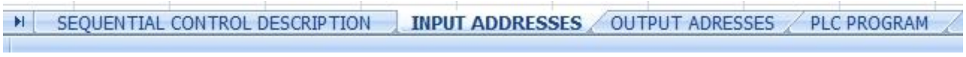


Fig. 6 Parts of automatic PLC programming by Excel

The first part is the sheet for describing the actuator movement according to some sequential control. The second part consists of two sheets for generating input and output addresses for sensor and command signals. The last sheet is used for generating PLC commands and program.

In the “SEQUENTIAL CONTROL DESCRIPTION” sheet, the operater describes the actuator movement according to a sequential control system. By recognizing the characters between “C” and signs “+” or “-”, the program detects the actuator number, and after signs “+” or “-” the sensor number. After that, the program prints the input and output addresses for sensors and command signals in the “INPUT ADDRESSES” and “OUTPUT ADDRESSES” sheets. Finally, the PLC commands and the entire program are generated in the “PLC PROGRAM” sheet.

When the generation of the input and output addresses and the PLC program sheet are completed by Excel, it is very easy to copy and paste those sheets to adequate software for PLC programming, such as FST or CoDeSys for FESTO PLCs, SIMATIC for SIMENS PLCs or any other software for programming the various PLCs. It is only important to use an adequate designation for input and output addresses depending on the PLC type.

6. EXAMPLE OF THE AUTOMATIC GENERATION OF A PLC PROGRAM BY MICROSOFT OFFICE EXCEL

As an example of the automatic generation of a PLC program by Excel, the problem of programming the movement of four pneumatic actuators in a sequential control system with single and parallel branches is considered. Actuators C_1 and C_4 have two end positions and actuators C_2 and C_3 have more middle positions to stop.

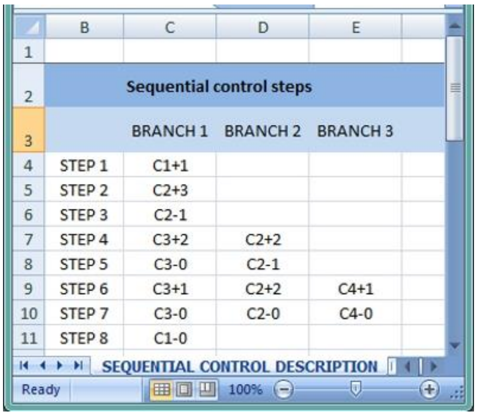
The movement of the actuators is as follows:

Step 1	**Step 2**	**Step 3**	**Step 4**	**Step 5**	**Step 6**	**Step 7**	**Step 8**
C_1+_1	C_2+_3	C_2-_1	C_3+_2	C_3-_0	C_3+_1	C_3-_0	C_1-_0
			C_2+_2	C_2-_1	C_2+_2	C_2-_0	
					C_4+_1	C_4-_0	

In this example, the four pneumatic actuators move in eight steps, where from step 1 to step 3 there is only one branch, from step 4 to step 5 there are two branches, and from step 6 to step 7 there are three sequential control branches. At the end of the sequential control system there is only one branch with step 8.

The pneumatic actuator type does not influence the PLC program because in this paper all of them need two command signals, due to the same type of the 5/2 or 5/3-way bistable direction valve for supplying them. The actuator type only affects the pneumatic scheme.

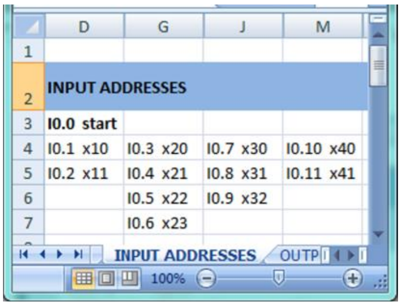
The first step in the PLC automatic programming by Excel is to fill the "SEQUENTIAL CONTROL DESCRIPTION" sheet, in a way that was explained in Chapter 5, Fig. 7.



	B	C	D	E
1				
2		Sequential control steps		
3		BRANCH 1	BRANCH 2	BRANCH 3
4	STEP 1	C1+1		
5	STEP 2	C2+3		
6	STEP 3	C2-1		
7	STEP 4	C3+2	C2+2	
8	STEP 5	C3-0	C2-1	
9	STEP 6	C3+1	C2+2	C4+1
10	STEP 7	C3-0	C2-0	C4-0
11	STEP 8	C1-0		

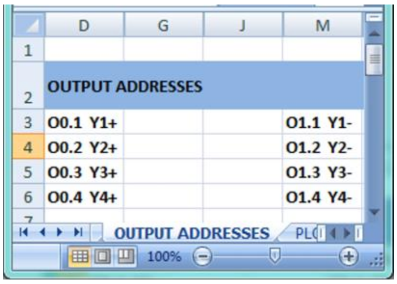
Fig. 7 "SEQUENTIAL CONTROL DESCRIPTION" sheet in Excel

By processing the characters from the "SEQUENTIAL CONTROL DESCRIPTION" sheet, Excel prints the input and output addresses for sensors and command signals in the "INPUT ADDRESSES", Fig. 8, and "OUTPUT ADDRESSES" sheets, Fig. 9.



	D	G	J	M
1				
2	INPUT ADDRESSES			
3	I0.0 start			
4	I0.1 x10	I0.3 x20	I0.7 x30	I0.10 x40
5	I0.2 x11	I0.4 x21	I0.8 x31	I0.11 x41
6		I0.5 x22	I0.9 x32	
7		I0.6 x23		

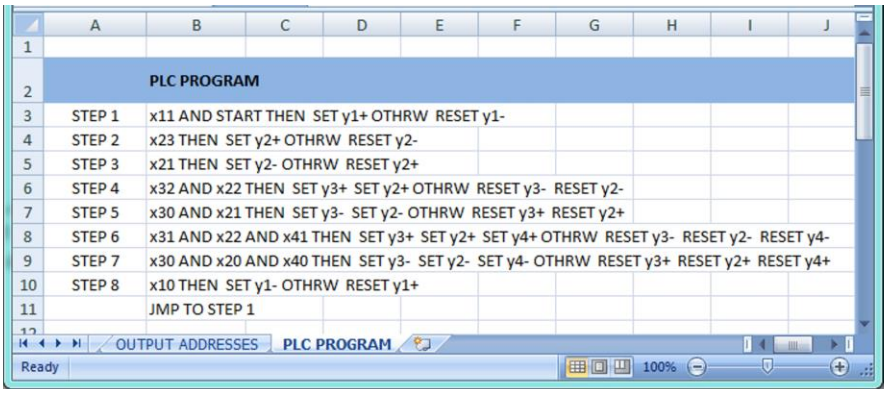
Fig. 8 "INPUT ADDRESSES" sheet in Excel



	D	G	J	M
1				
2	OUTPUT ADDRESSES			
3	O0.1 Y1+			O1.1 Y1-
4	O0.2 Y2+			O1.2 Y2-
5	O0.3 Y3+			O1.3 Y3-
6	O0.4 Y4+			O1.4 Y4-

Fig. 9 "OUTPUT ADDRESSES" sheet in Excel

Based on the input and output addresses for sensors and command signals from the "INPUT ADDRESSES" and "OUTPUT ADDRESSES" sheets, as well as the description of the movement of actuators from the "SEQUENTIAL CONTROL DESCRIPTION" sheet, and processing the condition for the execution of the next step of sequential control, the "PLC PROGRAM" sheet and the PLC commands and program are generated, Fig. 10.



	A	B
1		
2		PLC PROGRAM
3	STEP 1	x11 AND START THEN SET y1+ OTHRW RESET y1-
4	STEP 2	x23 THEN SET y2+ OTHRW RESET y2-
5	STEP 3	x21 THEN SET y2- OTHRW RESET y2+
6	STEP 4	x32 AND x22 THEN SET y3+ SET y2+ OTHRW RESET y3- RESET y2-
7	STEP 5	x30 AND x21 THEN SET y3- SET y2- OTHRW RESET y3+ RESET y2+
8	STEP 6	x31 AND x22 AND x41 THEN SET y3+ SET y2+ SET y4+ OTHRW RESET y3- RESET y2- RESET y4-
9	STEP 7	x30 AND x20 AND x40 THEN SET y3- SET y2- SET y4- OTHRW RESET y3+ RESET y2+ RESET y4+
10	STEP 8	x10 THEN SET y1- OTHRW RESET y1+
11		JMP TO STEP 1

Fig. 10 "PLC PROGRAM" sheet in Excel

7. Conclusion

The automatic generation of PLC commands and the entire program enables the operator to easily describe the movement of the actuators according to a certain established technological process in the way of sequential control by characteristic encoding. This way of programming PLCs is especially suitable for pneumatic systems. It enables the generation of a PLC program for a sequential control system with a single as well as parallel branches, by recognizing the conditions for the execution of individual steps. This paper presents implementation of the automatic generation of a PLC program in Microsoft Office Excel. This manner of PLC programming by Excel offers many advantages and is available on various devices such as PCs, tablets and smart mobile phones with the Windows operating systems, macOS, Android and iOS.

REFERENCES

1. Laughton, M.A., Warne, D.J., 2002, *Electrical engineer's reference book*, Newnes, Oxford, England, 1504 p.
2. Thakur, N., Hooda, M., 2016, *A review paper on PLC & its applications in robotics and automation*, International Journal of Innovative Research in Computer and Communication Engineering, 4(4), pp. 209-214.
3. Dworzak, L., Mikulczyński, T., 2009, *Synthesis of sequential control algorithms for pneumatic drives controlled by monostable valves,* Archives of Foundry Engineering, 9(3), pp. 35-40.
4. Malayappan, S., Raj, K.A., Arunachalam, S.S., Venugopal, S., Ramalingam, D., 2009, *Design of A sequential control circuit for an industrial robot using cascading method*, Proceedings of the International Conference on Man-Machine Systems (ICoMMS), Penang, Malaysia, pp. 3B3-1-3B3-5.
5. Mroz, P, Brol, S., 2017, *Sequential control strategy of pneumatic-hydraulic drive,* Proceedings of The Institute of Vehicles, 2(111), pp. 95-103.
6. Bayoumi, M.S., 2014, *Novel method for designing a sequential logic controller with intermediate stop of actuators,* International Journal of Computer and Information Technology, 3(3), pp. 643-650.
7. Hasdemir, I.T., Kurtulan. S., 2006, *Automatic PLC code generation using MATLAB*, Elsevier IFAC Proceedings, 39(17), pp. 131-136.
8. Salunke, R., Vikhe, P., Sarode, T., 2013, *Implementation of automatic PLC code from MATLAB simulation model using B&R automation target for simulink*, Elsevier, Int. Conf. on Control, Communication and Power Engineering, pp. 390-395.
9. Guttel, K., Weber, P., Fay, A., 2008, *Automatic generation of PLC code beyond the nominal sequence,* 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, pp. 1277-1284.
10. Schumacher, F., Schrock, S., Fay, A., 2013, *Tool support for an automatic transformation of GRAFCET specifications into IEC 61131-3 control code*, 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, pp. 1-4.
11. Schumacher, F., Fay, A., 2014, *Formal representation of GRAFCET to automatically generate control code*, Control Engineering Practice, 33, pp. 84-93.
12. Julius, R., Schurenberg, M., Schumacher, F., Fay, A., 2017, *Transformation of GRAFCET to PLC code including hierarchical structures*, Control Engineering Practice, 64, pp. 173-194.
13. Wcislik, M., Suchenia, K., Laskawski, M., 2015, *Programming of sequential control systems using functional block diagram language*, Elsevier IFAC Proceedings, 48(4), pp. 330-335.
14. Wcislik, M., Suchenia, K., Laskawski, M., 2016, *Method of programming of sequential control systems using LabVIEW environment*, Elsevier IFAC Proceedings, 49(25), pp. 476-481.
15. Adiego, B.F., Darvas, D., Blanco, E., Jean-Charles, T, Bliudze, S., Blech, J.O., Suarez, V.M.G., 2015, *Applying model checking to industrial-sized PLC programs,* IEEE Transactions on Industrial Informatics, 11(6), pp. 1400-1410.
16. Qamsane, Y., Hamlaouiy, M., Tajer, A., Philippot, A., 2017, *A model-based transformation method to design PLC-based control of discrete automated manufacturing systems,* 4th International Conference on Automation, Control Engineering and Computer Science (ACECS - 2017), 19, pp. 4-11.
17. Blagojević, V. Ranđelović, S., Milanović, S., 2018, *Automatic generation of PLC programs for pneumatic actuators sequential control with two end positions*, Proc. XIV International SAUM Conference on on Systems, Automatic Control and Measurements, Niš, Serbia, CD.
18. Bryan, L.A., Bryan, E.A., 1997, *Programmable controllers- theory and implementation,* Industrial Text & Video Company, Atlanta, 202 p.