GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

The journal is indexed in
EBSCO

ISSN 2545-4803 on line
DOI: 10.46763/BJAMI

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI)

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS

YEAR 2023

VOLUME VI, Number 2

AIMS AND SCOPE:
BJAMI publishes original research articles in the areas of applied mathematics and informatics.

Topics:

1. Computer science;
2. Computer and software engineering;
3. Information technology;
4. Computer security;
5. Electrical engineering;
6. Telecommunication;
7. Mathematics and its applications;
8. Articles of interdisciplinary of computer and information sciences with education, economics, environmental, health, and engineering.

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS
(BJAMI), Vol 6

ISSN 2545-4803 on line
Vol. 6, No. 2, Year 2023

EDITORIAL BOARD

CONTENT

Balkan Journal of Applied Mathematics and Informatics Online ISSN 2545-4083
Volume VI Number 2 Year 2023 UDC: 681.51:004.453

PROGRAMMING APLC CONTROLLER WITH A LADDER DIAGRAM

DARKO BOGATINOV AND SASO GELEV

Abstract. In this paper one industrial process and the automation of that industrial process are considered. In automation, the most important part is the controller. The controller is actually the brain of the process, and it regulates the switching on and off of all the elements involved in the process itself. The controllers that are used in industrial automation are called programmable logic controllers (PLC controllers). With an example it is shown how the PLC controller can be programmed with a Ladder Diagram. PLC programming involves creating a set of instructions that govern the behavior of a PLC to control machinery and processes. These instructions are written using specialized software tools and are executed in a sequential manner. The primary objectives of PLC programming are to ensure the safe and efficient operation of automated systems, maintain process accuracy, and facilitate quick responses to changing conditions.

1. Introduction

Industry automation becomes a spacious field in manufacturing which had an important role in an extensive range of industries beyond manufacturing [4,5].

Most SCADA systems use a programmable logic controller (PLC – Programmable Logic Controller) which is programmed and is responsible for managing the process.

EC 1131-3 is a global standard for control programming languages in PLCs. PLC programming languages are divided into two categories: text languages and graphic languages. [1]

Programmable logic controllers can be programmed in five ways, which are the following:

- LD - (Ladder Diagram)
- SFC - (Sequential Function Charts)
- FBD - (Function Block Diagram)
- ST - (Structured Text)
- IL (Instruction List)

Ladder diagrams are the most used way to program PLC devices as this way best corresponds to the physical characteristics of the devices.

The control of the PLC device is performed in two steps:

1. It reads the inputs and outputs of the PLC device
2. It reads the program that has been defined and acted accordingly, that is, it gives an output from the set inputs.

Keywords. APLC Controller, ladder diagram, industrial process, programming.

2. Programming of a PLC controller with LD

LD - (Ladder Diagram) is the most used programming language for programming PLC devices and this is due to the good visualization that this programming language possesses. Also, this way of programming has the easiest readability compared to other ways of programming PLC devices. It is simple to learn and understand this way of programming because the symbols it uses are already familiar to users from electrical schematics.

LD programming is actually the addition of appropriate symbols, and the name ladder diagram itself is derived from the fact that the programming code itself resembles a ladder. The primary use of this programming language is in bit logic, the inputs and outputs of logic gates.

The AND, OR, and NOT logic are considered the basic building block logic for designing the complicated logic to decide the output status [2]

PLCOpen is the organization that first started using these ladder diagrams and introduced a standard for the use of these ladder diagrams called IEC 1131-3. The appearance of a ladder diagram is shown in Figure 1.

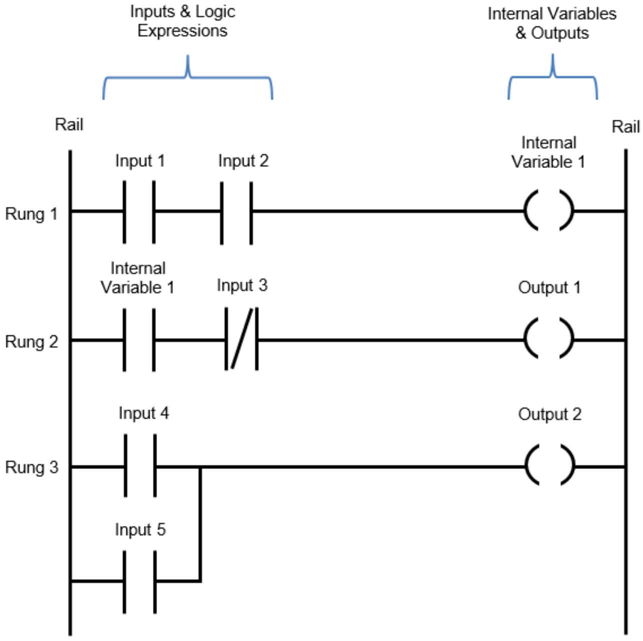


Figure 1. *Layout of LD - (Ladder Diagram)*

Before starting to program, the PLC device using a LD - (Ladder Diagram), it is necessary to first determine what the input is and the output of the PLC device are.

In the following we consider an example in which one tank will be filled automatically with liquids from two pumps. After filling, it is necessary to turn on a motor that has a switch that will mix the liquids for 7 seconds and then open the valve so that the liquid can continue to be used in production. The filling machine system consists of a prototype conveyor section, a filling section with sensors and a control section using PLC. The PLC control unit is utilized to reduce human work and increase the production with less time.[8]

First of all, the process needs to be defined and we need to see how it can be automated. By adding sensors to regulate the liquid level in the tank, the liquid level can be regulated. The pumps need to be connected to the PLC device so that their opening and closing can be controlled using the PLC device. The motor that drives the fin is also connected to the PLC device so in that way the PLC can regulate the work of the fin. Finally, the valve that should release the liquid from the tank should also be connected to the PLC device and this operation can be regulated.

Figure 2 shows the diagram and the way in which the used elements should be placed.

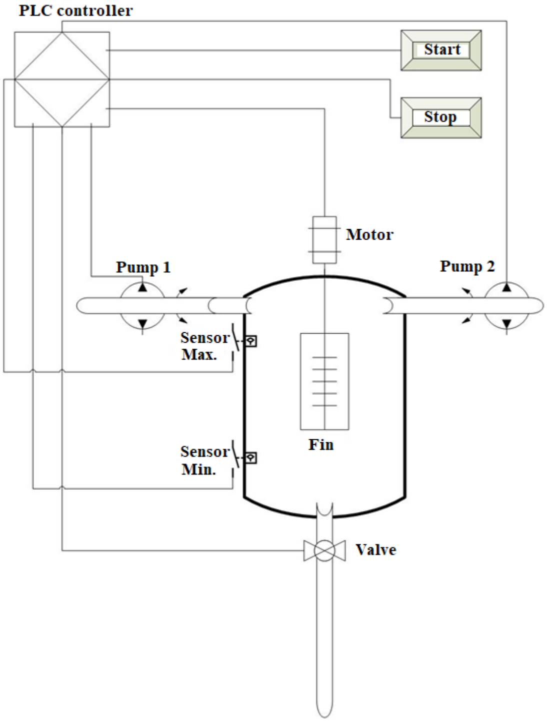


Figure 2. *Illustration of an example of an industrial process.*

The working principle is as follows:

When the sensor placed at the bottom of the tank Sensor Min. detects that the liquid level is low, it gives a signal to the PLC controller. The sensor is actually an open contact that, when it detects liquid, closes and the sensor sends a signal to the PLC device.

The PLC controller turns on the two pumps that refill the tank with liquid. When the liquid level reaches the sensor that is placed on top of the tank Sensor Max., the PLC controller turns off the pumps.

After the tank is filled and the pumps are turned off, the motor to which the fin is connected is turned on, which has the task of mixing the two liquids. After a defined time, when the liquids are sufficiently combined, the valve is opened, which releases the liquid from the tank that is used in further processes. The process is repeated, that is, when the liquid level reaches a minimum again, the pumps are turned on and the process is repeated again. At any stage of the process, if the user presses the STOP button, the process should be terminated.

Figure 3 presents the life cycle of the process.

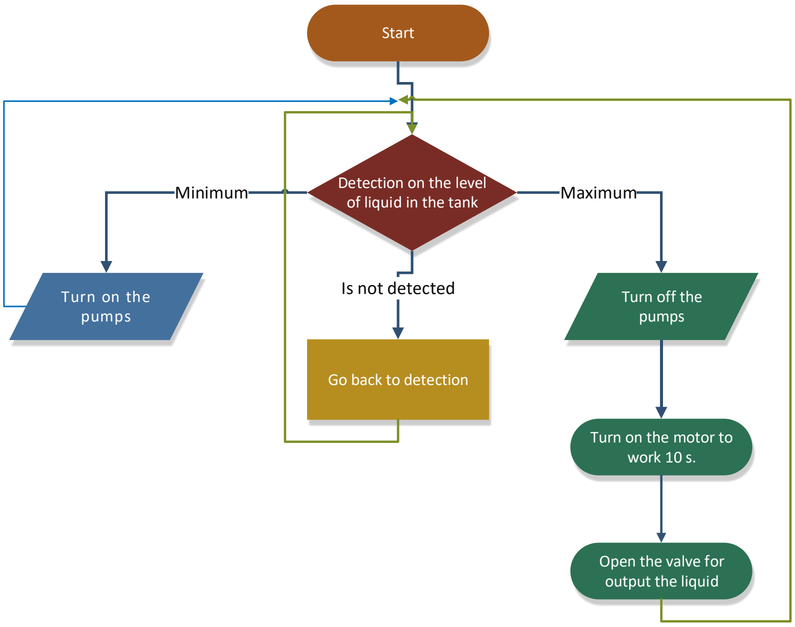

Start
Detection on the level of liquid in the tank
Minimum
Maximum
Turn on the pumps
Is not detected
Turn off the pumps
Go back to detection
Turn on the motor to work 10 s.
Open the valve for output the liquid

Figure 3. *Process description diagram*

The first thing that should be done before we start programming the PLC device is to determine what the input and the output of the process itself are. Specifically, in our example, the inputs are the following components:

- Sensor minimum
- Sensor maximum
- STOP button

The outputs are the following components:

- Pump 1
- Pump 2
- Motor
- Valve

Figure 4 presents the input and output signals of our PLC controller.

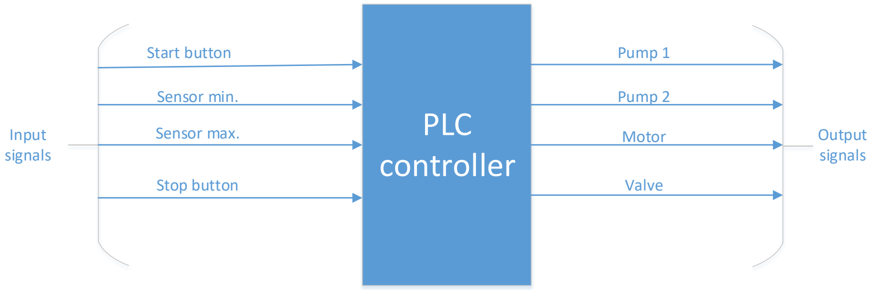


Figure 4. *Input and output signals of the PLC controller.*

When the PLC device is programmed using a Ladder Diagram, the PLC device actually reads the commands that are given from left to right. On the left side of the Ladder Diagram are the inputs and on the right side are the outputs of the PLC device.

During programming, symbols for on and off contact are used, which are shown in figure 5, on the left of the picture is the on contact and on the right is the symbol for an off contact.

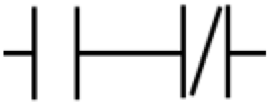


Figure 5. *Symbols for on and off contact in LD.*

The sensor actually represents a contact that is turned off, when it detects that the liquid level is up to the place where the contact is placed, it switches from off to on contact.

The START and STOP buttons are also off contact, when the user presses the button, the contact is on.

The outputs of the PLC device during programming using the LD (Ladder Diagram) are displayed by adding the symbol shown in Figure 6.

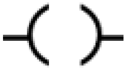


Figure 6. *Symbol for output in LD.*

The programming of the example and the simulation are done in the program Machine Expert – Basic; this is a program that was created by the manufacturer Schneider-electric and it is used for programming and simulation of PLC devices. Each manufacturer has its own program in which the programming of the PLC device is carried out.

The program Machine Expert - Basic is a program that is free to use and its use is not charged additionally.

For programming of the mentioned example, the type of PLC device TM221ME16R from the manufacturer Schneider-electric is used. This PLC device has 8 digital inputs and 8 digital outputs. For the example 4 inputs and 4 outputs are needed so that this device meets our needs. First of all, before we start programming the PLC device, we need to define the inputs and outputs that are used.

In our case, there are 4 inputs, which are:

- The signal from the START button
- The signal from the STOP button
- The signal from the sensor placed at the bottom of the tank sensor min.
- The signal from the sensor placed on top of the tank sensor max.

If the start and stop buttons are not pressed, they give a value of 0; that means there is an open contact, when the user presses the buttons, then they have a value of 1, the contact is closed.

The sensors work in a similar way if the sensor does not detect the presence of liquid; if the liquid level is below the level where the sensor is placed, it sends 0 to the PLC device; when the liquid level is at the level measured by the sensor, it sends 1 to the PLC device.

In the configuration section in the program, we need to add the inputs and outputs of the PLC device. As already mentioned, the selected PLC device TM221ME16R has 8 inputs and 8 outputs, the inputs have an address from I0.0 to I0.7 and the outputs of the device have an address from Q0.0 to Q0.7. When programming the PLC device for our example, the inputs and outputs are defined as follows:

Inputs:

I0.0 – Input signal from the start button (KOPCE_START)

I0.1 – Input signal from the stop button (KOPCE_STOP)

I0.2 – Input signal from the bottom sensor (SENZOR_MINIMUM)

I0.3 Input signal from the tank top sensor (SENZOR_MAXIMUM)

Outputs:

Q0.0 – Output to which one of the pumps is connected (PUMPA_1)

Q0.1 - Output to which one of the pumps is connected (PUMPA_2)

Q0.2 – Output to which the motor that moves the fin is connected (MOTOR)

Q0.3 – Output to which the discharge valve is connected (VENTIL)

The next step is the programming of the PLC device, i.e., giving commands which output should be active depending on the inputs that are active on the PLC device. Programming is done with an LD - (Ladder Diagram). Before we start programming, it is necessary to define the truth table in order to define the dependence of the outputs from the input signals. In our example, there are 4 input signals, so the possible input combinations would be 2^4=16, so there are 16 input signal combinations.

0 = open contact

1 = closed contact

In our example, in order for the system to start working, the user needs to press the start button, the start input signal must be 1. Therefore, in the truth table, if the start input signal is 0, the program should not do anything, none of the outputs is active. In places where the input signal is 0, the output signals will have a value of 0 everywhere on the output signals. This is shown in the truth table in the rows from sequence number 1 to sequence number 8.

Table 1. *Truth table*

Input					Output			
Sequence number	Start button	Stop button	Sensor minimum	Sensor maximum	Pump 1	Pump 2	Motor	Valve
1	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	1	1	0	0	0	0
5	0	1	0	0	0	0	0	0
6	0	1	0	1	0	0	0	0
7	0	1	1	0	0	0	0	0
8	0	1	1	1	0	0	0	0
9	1	0	0	0	1	1	0	0
10	1	0	0	1	1	1	0	0
11	1	0	1	0	1	1	0	0
12 A	1	0	1	1	0	0	1	0
12 B	1	0	1	1	0	0	0	1
13	1	1	0	0	0	0	0	0
14	1	1	0	1	0	0	0	0
15	1	1	1	0	0	0	0	0
16	1	1	1	1	0	0	0	0

When the input signal KOPCE_START is equal to 1, it means that the user activated the START button; that means the contact is closed and the program should start working. First, the program should activate the two pumps that will fill the tank with liquid. The tank should be filled up until both input signals that detect the level are 1. That means the signal from both the minimum sensor and the maximum sensor should be 1.

This is shown in the truth table in rows 9, 10 and 11.

When the START button is active, i.e., it has a value of 1, and both minimum and maximum sensors detect liquid, i.e., they have a value of 1, it is necessary for the pumps to stop working and to activate the motor that will mix the liquid with a fin. The motor should be active for 10 seconds. This means that when the start has a value of 1 and the sensor minimum and sensor maximum inputs have a value of 1, it is necessary that the MOTOR output is active, that means it has a value of 1. This is shown in row 12A in the truth table.

After the mixer has mixed the liquid, i.e., after 10 seconds, the motor should be deactivated so that it becomes 0 and the output to which the valve is connected should be activated and it should become 1. This is shown in the truth table in column 12B.

In rows 13 to 16, the outputs should always be inactive, i.e., 0, because the STOP BUTTON input is equal to 1, which means that the user pressed the STOP button.

After the defining of the truth table, equations for each of the outputs need to be determined. The point of interest is in the situation when the output should be active, that is when it has the value 1. For the outputs Pump 1 and Pump 2, it can be noticed from the truth table that they have the same values, so it is enough to determine only the equation for either of the two outputs. The characteristic equation is determined using a K-map, in a K-map with 16 cells because there are 4 input variables that need to enter the outputs that have a value of 1.

Table 2. *Minimization with K-map for outputs Pump1 and Pump 2*

Sensor minimum / Sensor maximum

Stop / Start	00	01	11	10
00				
01				
11				
10	1	1		1

According to the assigned K-map, the characteristic equation for the outputs Pump 1 and Pump 2 according to the already assigned addresses of inputs and outputs would be the following:

- Pump 1 output characteristic equation:

$$Q0.0 = I0.0 * \overline{I0.1} * \overline{I0.2} + I0.0 * \overline{I0.1} * \overline{I0.3}$$

- Pump 2 output characteristic equation:

$$Q0.1 = I0.0 * \overline{I0.1} * \overline{I0.2} + I0.0 * \overline{I0.1} * \overline{I0.3}$$

In a similar way, the characteristic equations for the output that should activate the motor and for the output that should activate the valve are obtained. In the truth table it is shown that the two outputs have a value of 1 for the same combination of input signals; this is shown in the truth table in rows with sequence number 12A and 12B.

Table 2. *Minimization with K-map for outputs Motor and Valve*

Sensor minimum / Sensor maximum

Stop / Start	00	01	11	10
00				
01				
11				
10			1	

According to the K-map and according to the addressing of the input and output signals, we would get the following characteristic equations:

- Motor output characteristic equation:

$$Q0.3 = I0.0 * \overline{I0.1} * I0.2 * I0.3$$

- Valve output characteristic equation:

$$Q0.4 = I0.0 * \overline{I0.1} * I0.2 * I0.3$$

After determining the characteristic equations, we can start programming the PLC device using a Ladder Diagram. During programming, in fact, the diagram itself is divided into several parts or steps (that is why this way of programming PLC devices is called a ladder diagram). On the first ladder, the equations obtained for the outputs Pump 1 and Pump 2 are implemented and this implementation is shown in Figure 7 in part Rung0. When creating the Ladder Diagram, the input signals are located on the left side of the diagram and the outputs are located on the right side of the diagram. We have already mentioned the symbols for open and closed contact: an open contact means that the input signal has a value of 1 and a closed contact means that the input signal has a value of 0. In order to activate the output located on the right of the diagram, it is necessary to meet the conditions that have been set on the left side of the diagram, the inputs have the set values 0 or 1 on the diagram itself.

In our example, by pressing the start button, the pumps should turn on and fill the tank with liquid to the level to which the sensor that detects the maximum level is set. This is achieved by adding the appropriate symbols which are shown in the first step Figure 7.

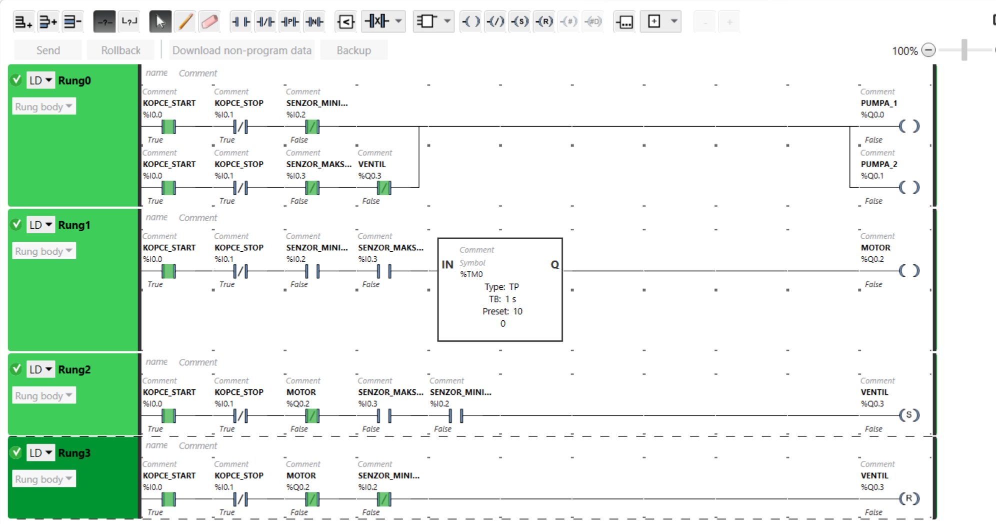


Figure 7. *Ladder Diagram for the example.*

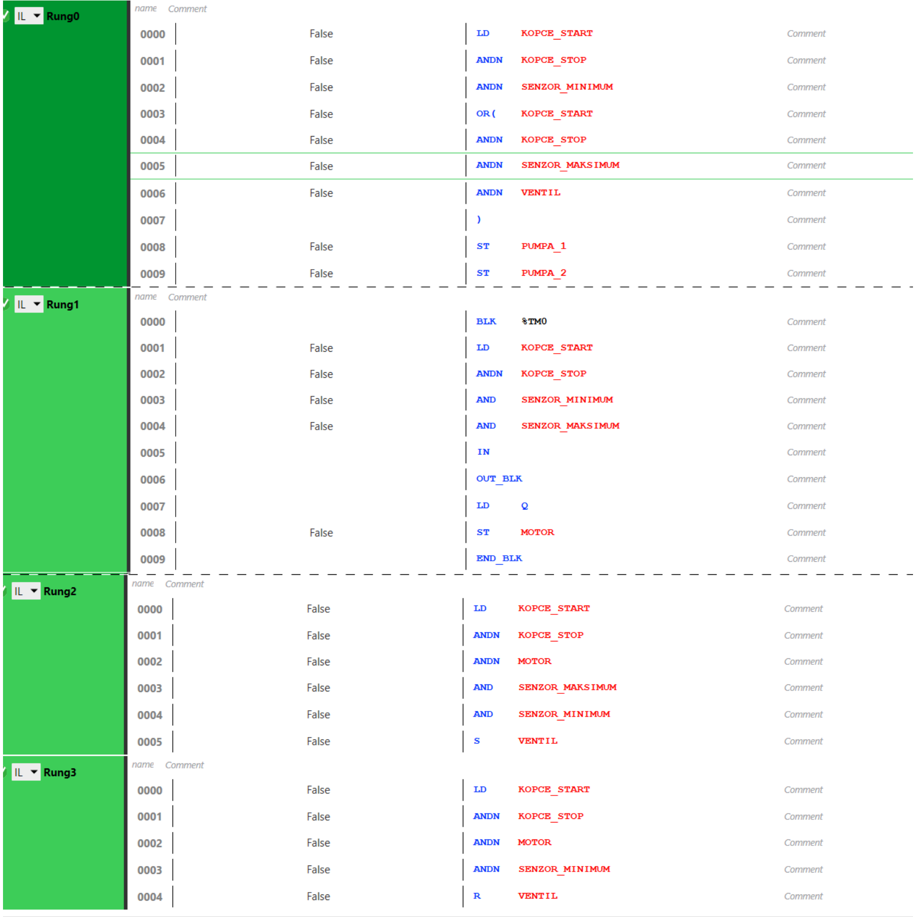
After the sensor is placed at the maximum level to detect that the liquid has reached the maximum level, it is necessary to turn on the motor to which the fin that mixes the liquid is connected. The fin needs to run for 10 seconds. Therefore, in the second step, the characteristic equation that is obtained for the MOTOR output is implemented, but also a timer that will activate the output for only 10 seconds is set. The implementation of the equation for motor output is implemented in the ladder diagram shown in Figure 7 in part Rung1.

After 10 seconds, the motor needs to stop working, it is necessary to open the valve from the tank and drain the liquid. This functionality is obtained by implementing the characteristic equation of the VALVE output that is obtained and additionally adds a condition that the valve output will become active when the MOTOR output is deactivated, i.e., after 10 seconds of activity on the MOTOR output. This functionality is obtained by implementing the third step shown in the ladder diagram shown in Figure 7 part Rung2.

The output valve should be active until the moment when the minimum sensor becomes 0, that is, until the tank has emptied the liquid to the minimum. This functionality is added by using the SET and RESET output functions. In the third step, the VALVE output has the value SET, which means that the input has a value of 1. In the fourth step, the condition is added that when the VALVE output is reset, that means the VALVE output changes from 1 to 0. The method of creating the fourth step is shown in Figure 7 part Rung3.

In this way the PLC controller is programmed to work properly with the conditions that need to be accomplished in the process.

The software Machine Expert - Basic has the possibility to convert the programming code performed with a Ladder Diagram into a program code with an IL (Instruction List). The program code with the instruction lists for the ladder diagram that is created during our programming is shown in Figure 8.



```
IL Rung0   name  Comment
0000  False   LD    KOPCE_START        Comment
0001  False   ANDN  KOPCE_STOP         Comment
0002  False   ANDN  SENZOR_MINIMUM     Comment
0003  False   OR(   KOPCE_START        Comment
0004  False   ANDN  KOPCE_STOP         Comment
0005  False   ANDN  SENZOR_MAKSIMUM    Comment
0006  False   ANDN  VENTIL             Comment
0007          )                        Comment
0008  False   ST    PUMPA_1            Comment
0009  False   ST    PUMPA_2            Comment

IL Rung1   name  Comment
0000          BLK   %TM0               Comment
0001  False   LD    KOPCE_START        Comment
0002  False   ANDN  KOPCE_STOP         Comment
0003  False   AND   SENZOR_MINIMUM     Comment
0004  False   AND   SENZOR_MAKSIMUM    Comment
0005          IN                       Comment
0006          OUT_BLK                  Comment
0007          LD    Q                  Comment
0008  False   ST    MOTOR              Comment
0009          END_BLK                  Comment

IL Rung2   name  Comment
0000  False   LD    KOPCE_START        Comment
0001  False   ANDN  KOPCE_STOP         Comment
0002  False   ANDN  MOTOR              Comment
0003  False   AND   SENZOR_MAKSIMUM    Comment
0004  False   AND   SENZOR_MINIMUM     Comment
0005  False   S     VENTIL             Comment

IL Rung3   name  Comment
0000  False   LD    KOPCE_START        Comment
0001  False   ANDN  KOPCE_STOP         Comment
0002  False   ANDN  MOTOR              Comment
0003  False   ANDN  SENZOR_MINIMUM     Comment
0004  False   R     VENTIL             Comment
```

Figure 8. *Instruction List code for the example*

3. Results and discussion

Programmable Logic Controllers (PLCs) are integral components in industrial automation and control systems, playing a pivotal role in managing and monitoring various processes across industries. PLC programming is a critical aspect of industrial automation and control systems. It enables the automation of various processes in manufacturing, energy, and other industries, leading to increased efficiency, reduced human error, and improved safety. Properly programmed PLCs enhance safety by ensuring that machinery and processes operate within defined safety limits. Emergency shutdowns and fault detection can be integrated into the program to prevent accidents. PLC programs can be easily scaled up or down to accommodate changes in the production volume or process requirements. This scalability makes PLCs a cost-effective choice for many industries. PLC programming is at the heart of modern industrial automation, and mastering it is crucial for ensuring efficient and safe industrial processes. As technology continues to advance, PLCs will likely play an even more significant role in shaping the future of industrial automation.

References

[1] *NORAZILA BINTI MD POSDZI, TAN EE CHIN:* "PROGRAMMABLE LOGIC CONTROLLER (PLC): PROGRAMMING & APPLICATIONS" Publisher, Politeknik Ungku OmarJalan Raja Musa Mahadi31400 Ipoh, Perak

[2] *Yasser Elsayed:* DESIGNING LOGIC GATES IN PLC SIMULATOR Home PLC Projects Designing Logic Gates in PLC, December 2021

[3] *W. Bolton:* Ladder and Functional Block Programming, CHAPTER 11

[4] *A. Dakre, J. Sayed2 , E. Thorat*,"Implementation of Bottle Filling and Capping using PLC with SCADA", International Research Journal of Engineering and Technology, Vol.2, Dec2015

[5] *T. PRATAP, S. SINGH*, "Automation of Bottle Filling System in Industries using PLC and SCADA", International Research Journal of Engineering and Technology, Vol.3, Sep.2016

[6] *Shaukat N.*, PLC Based Automatic Liquid Filling Processdz, IEEE Multi Topic Conference, 2002.

[7] *S. Ould Biha*, "A Formal Semantics of PLC Programs in Coq," 2011 IEEE 35th Annual Computer Software and Applications Conference, Munich, Germany, 2011, pp. 118-127, doi: 10.1109/COMPSAC.2011.23.

[8] *D. Baladhandabany, S. Gowtham*,"PLC Based Automatic Liquid Filling System", International Journal of Computer Science and Mobile Computing , Vol.4, March 2015 .

Darko Bogatinov
Goce Delcev University, Stip
Faculty of Electrical Engineering, Krste Misirkov 10-A
2000 Stip, Republic of North Macedonia
E-mail address: darko.bogatinov@ugd.edu.mk

Saso Gelev
Goce Delcev University, Stip
Faculty of Electrical Engineering, Krste Misirkov 10-A
2000 Stip, Republic of North Macedonia
E-mail address: saso.gelev@ugd.edu.mk