IOP Conf. Series: Materials Science and Engineering **227** (2017) 012097 doi:10.1088/1757-899X/227/1/012097

An innovative approach for modeling and simulation of an automated industrial robotic arm operated electro-pneumatically

L Popa[1], **V Popa**[2]
[1]Department of Automation and Information Technologies, Transilvania University, Mihai Viteazul, No.5, 500036, Braşov, Romania
[2] Department of Defense Resources Management Studies , National Defence University, Mihai Viteazul, No. 160 , 500183, Braşov, Romania

E-mail: mluminita2001@yahoo.com

Abstract. The article is focused on modeling an automated industrial robotic arm operated electro-pneumatically and to simulate the robotic arm operation. It is used the graphic language FBD (Function Block Diagram) to program the robotic arm on Zelio Logic automation. The innovative modeling and simulation procedures are considered specific problems regarding the development of a new type of technical products in the field of robotics. Thus, were identified new applications of a Programmable Logic Controller (PLC) as a specialized computer performing control functions with a variety of high levels of complexit.

1.Introduction

French Standardisation Institute defines a robot as an “automatic polyvalent manipulator, able to achieve positioning and orientation of the pieces by a variable and programmable movement of the arms terminals, through devices”. American Institute of Robotics considers the robot as a reprogrammable and multifunctional controller for moving objects on set trajectories in achieving concrete tasks. Handling robotic arm electro-pneumatic actuated is used successfully in areas that use materials such as wood, plastic, metal (sheet metal) and glass.

2.The robotic system

A robotic system usually consists of a mechanical manipulator, an end-effector, a based controller - microprocessor, a computer and other devices [1]. A mechanical manipulator is designed from several links connected by joints.

The manipulator’s link is fixed in the ground, while following link is given as the finishing link. The current number of joints usually equals the number of degrees of freedom.

In this case, the manipulator could be controlled easily. The end effector is a device attached to the end mechanical grip manipulator links. Usually called industrial robots, these moving automated complex devices, used in manufacturing processes are an emerging industrial engineering chapter, occurred in the merger of automation and electrical drives.

3. The manipulator

A manipulator is a pseudo - robot with a simple control system through a rigid schedule, with a high application [2]. The manipulator role is to manipulate pieces (grip, separation) and move them to the work environment. If the manipulatorit is driven by a system of controls sufficiently developed (handling + performance), this manipulator becomes an industrial robot that acts, (in addition to handling parts and moving specific tools of a technological process) and may involve complex operations such as welding, painting, assembly, drilling. The most common industrial robot is the robotic arm with six degrees of freedom, easily reprogrammable and easy to include in assembly lines. To achieve this mechanism, each segment of the arm is actuated by rotating or linear motors, all coordinated to compose the desired movement to the enforcement end of the arm.

The kinematic robot coordination is performed by a controller, which has a hardware and software reprogrammable sub-system. The advantage of this automation is the possibility that the company to modify rapidly the production (almost without interrupting the flow / working robot line). A robotic arm is mechanical programmable assembly with functions similar to a human arm. The robotic arm must fulfil a series of tasks involving handling and transporting objects or using equipment and tools. Prehension and handling plays an important role in the industry because the most industrial robots are performing gripping parts tasks to handling / transfer from an initial position to a final position in an action-required trough a robotic process technology. The gripping process has several stages: approach to the object stage (the robotic arm is approaching to the object), hand orientation stage (the hand is positioned in the configuration needed to grab the object), object gripping stage (thumbs are close until they make contact with the object), pretension validation stage (verifying that all fingers are in contact with the object).

A typical handling action is to catch the object, moving it from an initial position and orientation in a different position and orientation target, avoiding contact with obstacles or interaction with another object. An important issue is the choice of the robotic handling clamping force, to exclude or minimize the risk of slipping.

3.1. Robotic arm orientation in 3D virtual environment using Solid Works software package

The robotic arm is able to do simultaneously hand orientation stage with approach to the object stage. This method is based on an decision algorithm based on data from the two sensors. The structure is designed in 3D environment using the Solid Works software package, and has taken into account all design parameters of the real model. In the virtual environment was simulated current position of the mechanical system and movements performed by the system. Computer Aided Design allows modification, analysis and optimization of a project. CAD software allows creating a 3D model of an object or set of objects and carrying out simulations and analysis. These simulations have behind well-established mathematical models, which take into account the structural properties of the represented virtually bodies. To study the movement of a robotic system and its behaviour under different working environment needs to be created that system will have to act, which involves certain costs. The situations in which robotic system to be tested is more, the amount invested in this study increases. Therefore, a less expensive option is the simulation of the entire process in a virtual environment because the virtual environment allows simulating an unlimited number of conditions. For this the robotic system should be introduced into the virtual environment.

You can see all assembled components forming the robotic arm in figure 1.

The final version of the presentation of the robotic arm is shown in figure 2.

The assembly of this figure consists of the following components and 3D parts: the 3D model of the pneumatic cylinder UNSD-20-100; the 3D model of the pneumatic-cylinder DMML-20-50; the 3D model of pneumatic rotary-cylinder DSR-16-180-P; the 3D model Vacuum Generator / Ventura head VAD 1/8; the assemble between the rotating cylinder DSR-16-180-P and the pneumatic cylinder DMML -20-50; the attachment of pneumatic cylinder DMML -20-50 to the pneumatic cylinder created in Stage 1; the attachment of the vacuum generator VAD 1.8, plus the supporting elements of this; the attaching the vacuum generator VAD 1/8, plus the supporting elements of this, to the assembly created in Stage 2. At the Assembly already created in previous stages is attaching a support base.

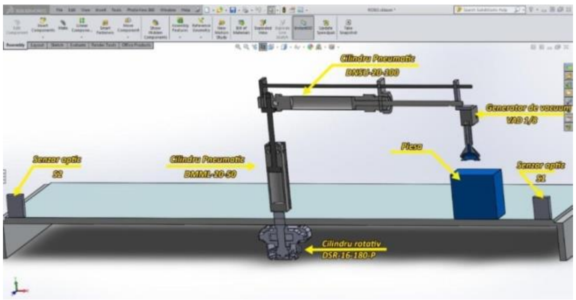

Figure 1. Sectional view of the robotic arm.

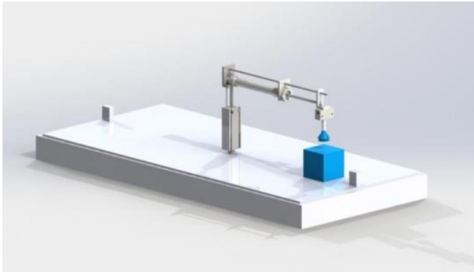

Figure 2. The final version of the presentation of the robotic arm.

3.2. The cinematic scheme of robotic arm

Structural analysis of the cinematic scheme is according to the coordinate system (Cartesian, cylindrical and spherical) in which is moving the robot's mechanical hand. The cinematic scheme of choosing process, is determined by structural and technological construction features of the machine or technological equipment that the robot serves and shape and dimensions of the manipulated part.

Our paper presents an industrial robot with movements of rotation and translation controlled by a programmable logic controller. Actuators are DC motors used for the modifications trajectory. For the pneumatically actuated robotic arm to become fully automatic, I used a PLC, which is essentially a microprocessor that has a friendly software programmer. The drawback of this would be its price.

The arm is made up entirely of Festo pneumatic cylinders that make 5 functions. To have fully control of pneumatic cylinders will be use a magnetic sensors, directly mounted on the cylinder channel. These sensors will notify the sensors position (retired or advanced). In addition to these directly mounted sensors on the cylinder, there are used two reflective photo sensors, to know when the cube is on the right arm and when it reached the destination. Because there are sensors on arm (plus two for the starting position and end position), we can know at any time where is the arm and object handling it.

Sensors errors will not occur, as in the case of human error. Some applications require only the arm to move the product along desired axes, while another robotic arm handles an object in several different axes.

The robotic arm can be easily implemented on a large scale in almost any industry where handling things. It has high accuracy and adjustable speed, so it can perform manipulation of objects as fast as you want. This robotic arm is equipped with two buttons besides the one to *start*, those are for *stop* and *restart*. The *stop* button helps if you want to stop the robotic arm in every stage, and the restart button helps you to bring it to its original position after you hit *stop* button, and when the arm is stocked.

Thanks to these features listed above, it is not necessary the staff training and the robotic arm could be used by anyone. The robot enables automatic identification of the of geometric shapes type triangle, square, circle using a video camera.

The command system can be done manually or using a programmable logic controller console or on a schedule. The robot can pick, sort and sit in different containers three different geometric shapes. The rotation is ensured by a continuous current motor type EVP-120 through a worm gear, the reducer which ensures amplification of the torque while reducing speed. Vertical translational movement is obtained from a DC motor UHT-50 via a belt, which also provides the flexible coupling.

The transformation of rotation movement into a translational movement is performed by a screw-nut assembly. The gripping device is of the electromagnetic type. This system is very useful in industry for each track selection or sorting scrap of non-conforming parts.

Stage 0: Or as I said I "The initial position of the arm". From this position, the robotic arm starts moving, upon notification of his right sensor.

As is shown in figure 3, the sensor confirms that a piece (the cube) is beside it. So, we will switch to Stage 1.Stage 1: Once the sensor (S1) has notified the presence of the piece (the cube), the pneumatic cylinderDNSU-20-100 is expanding, as is shown in figure 4.

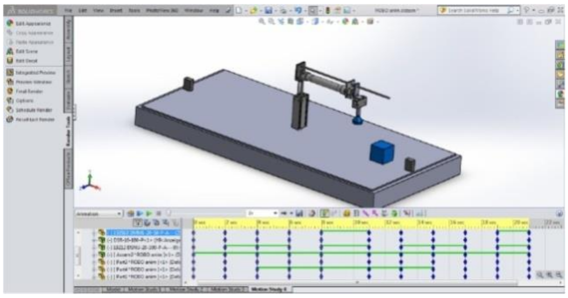

Figure 3. Stage 0 "The initial position of the arm". From this position, the robotic arm starts moving, upon notification of his right sensor.

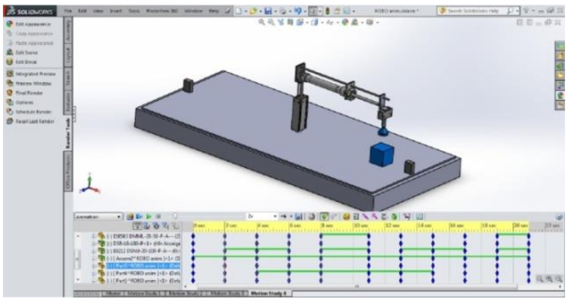

Figure 4. Stage 1. The pneumatic cylinder DNSU-20-100 is expanding.

Stage 2 At the moment the pneumatic cylinder DNSU is expanding, after this operation the pneumatic cylinder DMML -20-50 will descend, and the cup will act, figure 5. Stage 3: After Stage 2 was carried out, the next operation will be to lift the piece, figure 6. That the pneumatic cylinder DMML is extending / rise together with the piece.

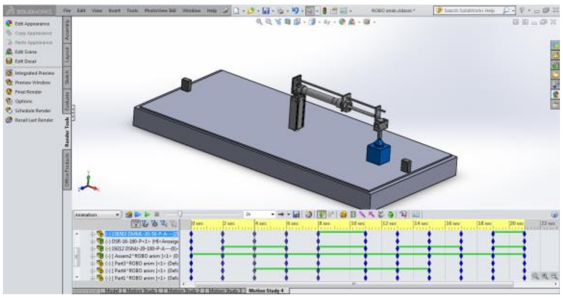

Figure 5. Stage 2. The pneumatic cylinder DMML-20-50 descends and the suction cup acts.

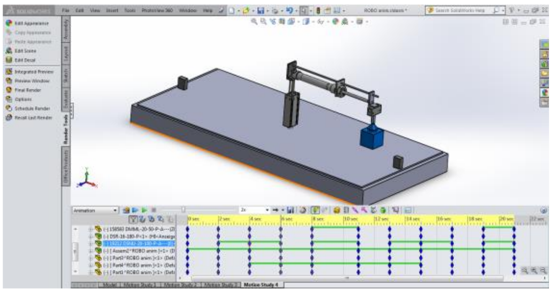

Figure 6. Stage 3. The pneumatic cylinder DMML is extending / rises together with the piece.

Stage 4: At this stage, the pneumatic cylinder DNSU retracts (figure 7), the suction cup remains engaged.
Stage 5: The rotary cylinder DSR-16-180 - P acts (figure 8)

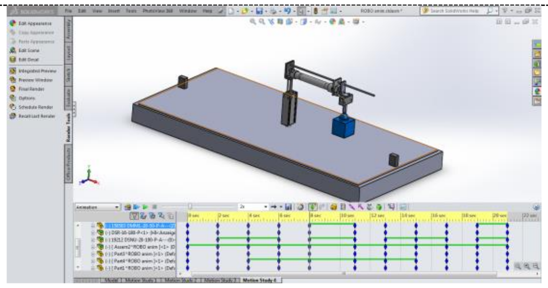

Figure 7. Stage 4. The pneumatic cylinder UNSD retracts.

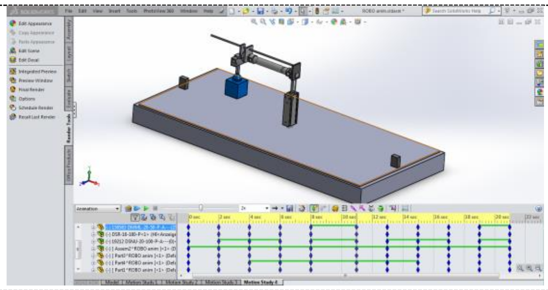

Figure 8. Stage 5. The rotary cylinder DSR-16-180 - P acts.

4. PLC programming

A new approach to automation systems is to use programmed logic and this category includes all systems that operate based on a recorded program in memory and implicit PLCs [3].

Most often used in practice there are automation systems containing both wired logic and programmed logic. Automated process control systems are a conglomeration of electronic devices that provides stability, accuracy and performance. Operating systems can have different implementations from power supplies to machine tools. Being a result of rapid technological progress, most complex operations were solved by connecting the system to PLCs and CPUs process. These PLCs (PLC - Programmability Logic Controller), besides connections to instrumentation and sensors in the automation process, should allow control of the whole process and to communicate to the operator the process states by visual signals and sounds and/or through a communications network to a local computer. These PLC features allow automation operation of the control unit to a high degree of fle xibility by changing and more convenient monitoring of basic parameters of the process. Each component of the process control system plays a role in line with its importance. For example, without a sensor, the PLCs could not know how to change the time parameters of the process (considered input parameters). In the automated systems, PLCs are the central part of control system or automation.

PLCs are structures for the management of industrial processes, the implementation of which was followed in as much as the elimination of wired logical structure and the replacing them with programmable logic structures. PLCs can be reprogrammed via PC, but it is possible to be programmed manually via programming console or, in case of PLC with reduced capacity, with buttons and displays set on the front of them. The intelligence a PLC is its ability to detect different types of sensors and input devices. Usually at a PLC, the most used features from the front panel are the buttons, the keys and other switches. The Schneider PLC is designed for use in small automation, comprising 10 to 40 inputs / outputs. This PLC can be used both in industrial applications (small machine tools, irrigation, pumping stations) and in the commercial applications (management of lighting, heating, air conditioning, etc.) Programming can be done:

- directly using the buttons on the relay;
- on a PC using "Zelio Soft" and a programming cable, with the possibility of programming in FBD language (Function Block Diagram) or Ladder.

The following is a command application of robotic arm and it is simulated using Zelio Soft program. This software is used by the Zelio automatic machine.

4.1. Introduction to Zelio Soft software

When we open the Zelio Soft program, will appear the following window (figure 9):

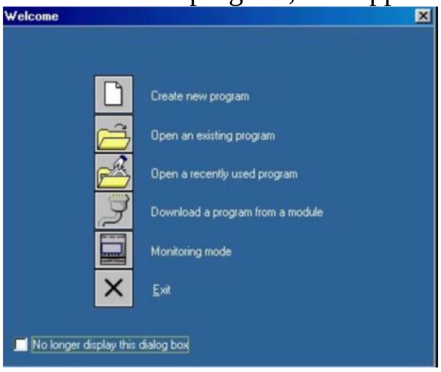

Figure 9. Zelio Soft opening window.

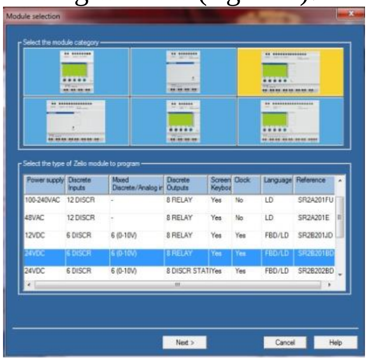

Figure 10. The window for choosing the type of PLC.

To create a new program [4], we can select the first option “Create a new program” or, if the program is already open, we select “New”. Once we passed this stage, will appear the following window, from where we choose the PLC type. We used type SR2 B201BD as can be seen in figure 10.

To move to the next stage, we press ***next***. After we have selected ***next*** window will appear (figure 11):

As you can see, in the figure 12 are two options Ladder or FBD. We have chosed the option FBD. Once, we have selected the desired option, we click *Next* and will appear the edit window.

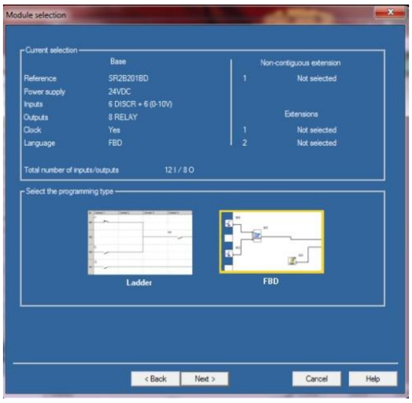

Figure 11. Choosing the PLC type.

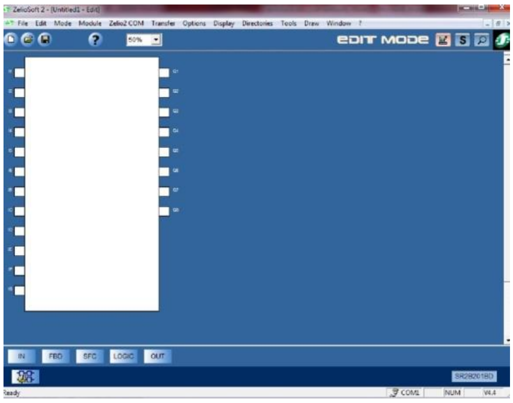

Figure 12. The edit window.

4.2. FBD –a programming language offering multiple possibilities

A Zelio Logic machine can be programmed in language FBD (Function Block Diagram), which is a graphical language that offers many possibilities.

We can also directly create graf- sites using SFC (Sequential Function Chart) - Grafcet (Graphe Fonctionnel de Commande Etapes) accessible from FBD.

A Zelio Logic machine Toolbar is represented in figure 13.

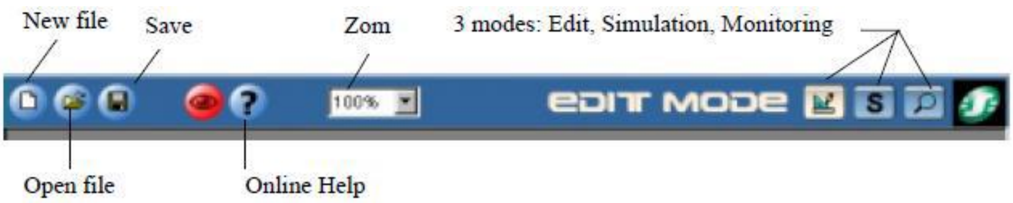

Figure 13. Figure a Zelio Logic machine Toolbar.

4.3. Entering a program in FBD

Once we have selected the type of machine (SR2B201BD) and FBD programming language, we are ready to start the application. To check again the type of used machine, we need to look at the bottom of the program, (figure 14).

Figure 14. Checking the type of used machine on the toolbar.

The program has three modes to choose: Edit mode (1), Simulation mode (2) and Monitoring mode (supervision) (3). These three modes are founded in the upper right corner of toolbar (figure 15).

IOP Conf. Series: Materials Science and Engineering **227** (2017) 012097 doi:10.1088/1757-899X/227/1/012097

Figure 15. The toolbar with the three modes of the program.

1. **Edit mode -** lets you to edit just like in the supervision mode.
2. **Simulation mode -** allows you to simulate the program before is transferred automatically.
3. **Monitoring mode -** allows you to visualize in real time the status of inputs and outputs.

4.4. Edit mode

After we passed the stage of automatic choice of type and language, it will see this editing spreadsheet. This spreadsheet shows the inputs (1), the outputs (3) of the automatic machine and the area reserved for programming blocks (2), (figure 16).

Figure 16. The editing spreadsheet.

Figure 17. The toolbar with the components blocks.

On the bottom of the toolbar can be seen the components blocks:
(1) Inputs (IN); (2) FBD functions (FBD); (3) The Grafcet functions / SFC; (4) The logic functions (LOGIC); (5) Outputs (OUT) (Figure 16).

To see the list of available blocks we could pass the mouse through shortcuts (functions, inputs, outputs) presented in figure 17, above. The list of available blocks is seen in figure 18.

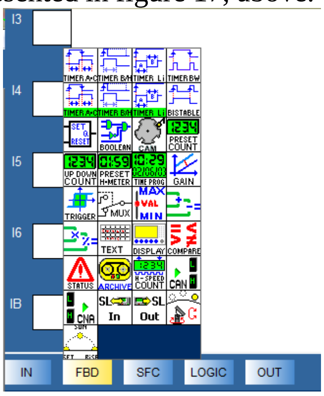

Figure 18. The list of available blocks.

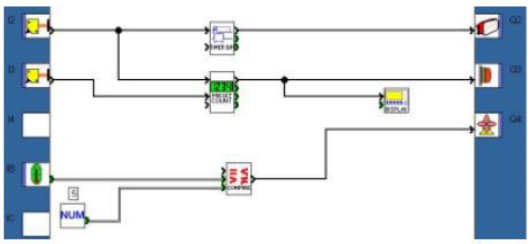

Figure 19. An example of using drag and drop.

Once we got here is very easy to choose the block that we need by using **drag** and **drop**. An example can be seen in figure 19.

4.5. A summary of the functions used in the program blocks

This software offers a multitude of blocks; we cannot present all of them.

To find out more information about the blocks, we can do the following: Double click on the block of interest in the open window and click the *question mark* (?).

4.6. The Transfer Program mode:

To send the program to Zelio Logic Automatic Machine, select "Transfer" from the menu, then click on *PC-> Module*. If the machine is connected to the PC, is in RUN mode and is impossible to transfer the program. To solve this issue, is necessary to switch from RUN mode to STOP mode (Transfer-> Stop modules). If the selected machine type is the same as the connected machine, the next window dialog appears (figure 20):

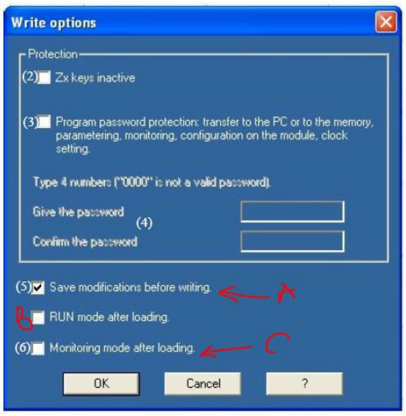

Figure 20. The program transfer.

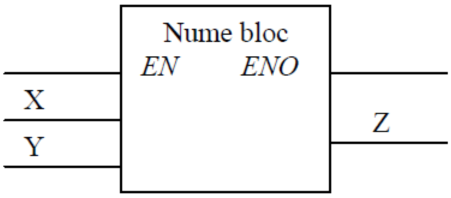

Figure 21. The functional block.

From this point we select the desired additional options, such as:
A. Saving changes before writing.
B. To operate in RUN mode, after loading the program.
C. To operate in Monitoring mode, after charging the program.

4.7. FBD programming language

FBD is a graphic language. It allows the programmer to build complex function blocks using existing libraries of programming environment.FBD program consists of basic functions blocks, connected by link lines. As the LAD program, FBD program runs from top to bottom and from left to right. Each block has a number of inputs and outputs.

The block has been represented by a rectangle. Inputs are on the left and the outputs on the right. A basic block performes a single processing action over the inputs.The function performed by a block, is written within.At the block entries there are linked the input variables and output variables of the blocks can be connected to PLC outputs or inputs of other blocks.The input variables type must match the required type of block entry.The block output has the same type as the block input.Every block has, in addition to entries on which are conducted operations, X respectively Y, an input named EN and an output named ENO, in addition to output Z.When EN is FALSE, we not operate on input data and ENO output is FALSE. When EN is TRUE, the block becomes operational and ENO output goes in TRUE state. If an error occurs during operation, the ENO GOES to FALSE state (figure 21).

The main blocks can be divided into the following categories:
- Standard blocks that correspond to the standard operators of the language STL (Statement List programming);
- Special units, implemented through complex procedures.

Standard blocks are:
- Blocks of data manipulation;
- Blocks for boolean operations (AND, OR, XOR, etc. Figure 22);
- Arithmetic blocks (for performing elementary operations, addition, subtraction, multiplication and division);
- Comparison blocks.

Special blocks are:
- Blocks of data manipulation (SR and RS flip-flops and detection of increasing and decreasing fronts, multiplexers, random number generators);
- Counters, timers, signal processing blocks (hysteresis or Schmitt trigger, Regulating devices, integration devices, derivatives, etc.);
- Signal generators blocks (rectangular signal generators, signal modulation generators for duration;
- Mathematical blocks (absolute value calculation, the exponential function, logarithm, square root, trigonometric functions, etc.).

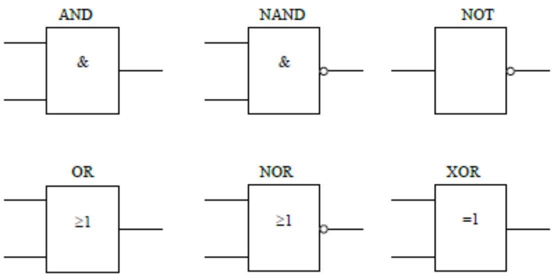

Figure 22. The standard blocks of language FBD for boolean operations.

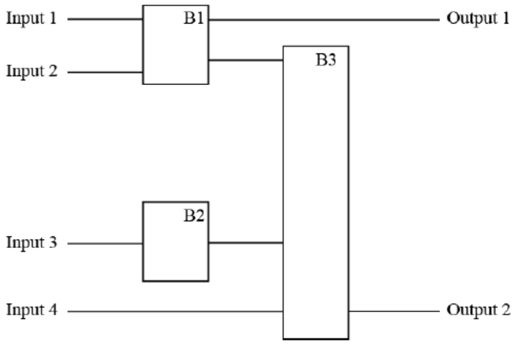

Figure 23. FBD layout program.

4.8. The principle of this program

As a principle of realization of this program we used the method of ”R-S flip-flops use for application programming starting with automation graph ”. Starting from the automation graph can be achieved programs for PLCs using blocks type flip-flop.The flip-flop is an elementary sequential block with two inputs: S - input setup; R - reset input and an output Q, which remembers the last order received.Most PLCs programming environments have implemented flip-flops R-S or S-R blocks (the first letter is dominant input that determines the condition if both inputs are active).Output logical equations are:

$$Q^{n+1} = S + \bar{R}Q^n \tag{1}$$

$$Q^{n+1} = \bar{R}(S + Q^n) \tag{2}$$

For the R-S flip-flop, which has been noted Q^n with the previous command state and with Q^{n+1}, the after command state. Transforming a program of the graph automation can be achieved using the R-S flip-flops. The following rules apply: Each variable attached to a stage is the output of a flip-flop; S(input setup) of corresponding flip-flop stage is the stage activation function; Input R of corresponding flip-flop stage is the stage to be activated when the stage is deactivated (figure 23).

If a stage has simultaneously *on* and *off* conditions, use of this method is not possible because one of the flip-flop inputs is dominant. For example, if the input R is dominant then the stage turns off although it also has activation conditions. One solution used to remove this drawback is the introduction of an additional stage to be switched off in a very short time by a timer, as will be seen in the following example.For example, for a sequence of three stages represented in figure 24, the FBD program is shown in figure 25. For this specification graph, transitions scroll functions and **enabled stages functions** are shown in relations (l) and (2). Entering the initial marking is done by setting flip-flops which represent the active stages through a variable of type *one shot*: INI, which is shown below (figure 25):

IOP Conf. Series: Materials Science and Engineering **227** (2017) 012097 doi:10.1088/1757-899X/227/1/012097

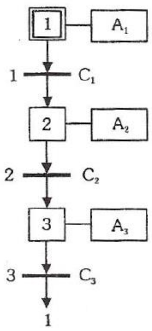

Figure 24. Unique three-step sequence.

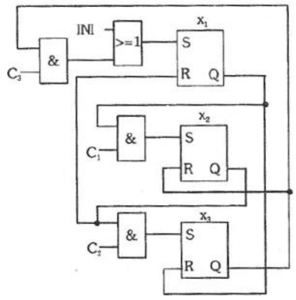

Figure 25. FBD language program for a sequence of three stages.

4.9. The final version of the program

The case in which we simply justify the caption so that it is as the same width as the graphic, is shown in figure 26a.

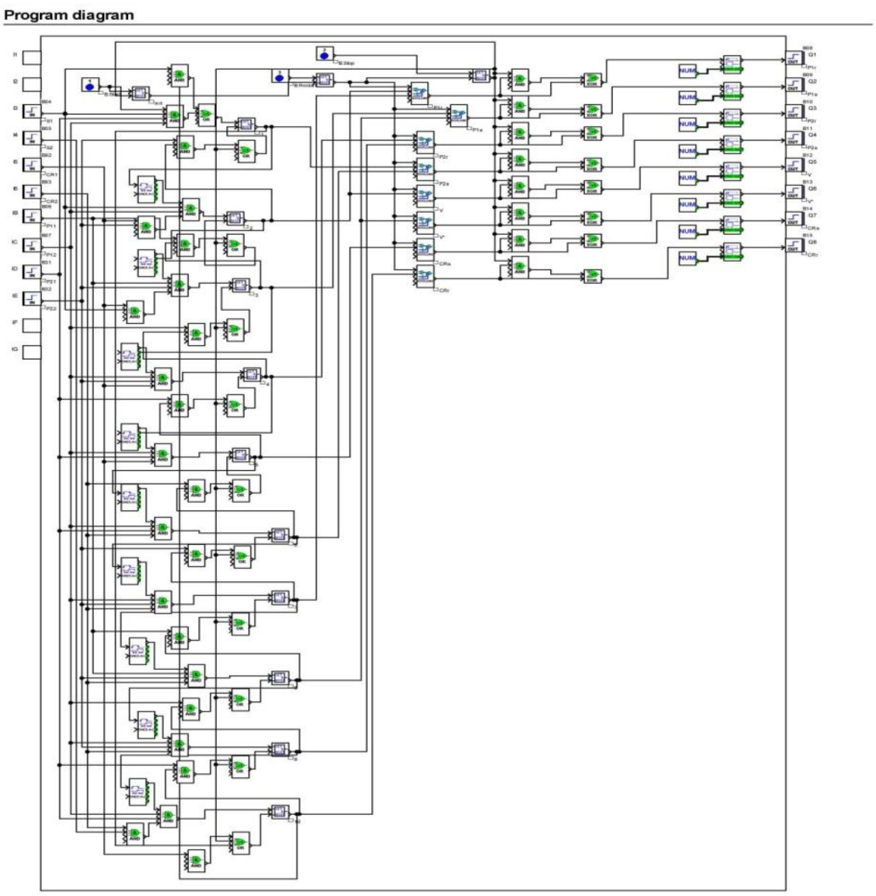

Figure 26a. The case in which we simply justify the caption so that it is as the same width as the graphic.

The PLC inputs/outputs and buttons are shown in figure 26b, 26c and 26d:

Input	No	Symbol	Function	Lock	Parameters	Comment	
I3	B04	IN	Discrete input	---	No parameters	S1	Initial position sensor
I4	B05	IN	Discrete input	---	No parameters	S2	End position sensor
I5	B92	IN	Discrete input	---	No parameters	CR1	Initial position sensor - - rotating cylinder
I6	B93	IN	Discrete input	---	No parameters	CR2	End position sensor - rotating cylinder
IB	B06	IN	Discrete input	---	No parameters	P11	Position sensor - piston withdrawal 1
IC	B07	IN	Discrete input	---	No parameters	P12	Position sensor - piston advancement 1
ID	B31	IN	Discrete input	---	No parameters	P21	Position sensor - piston withdrawal 2
IE	B32	IN	Discrete input	---	No parameters	P22	Position sensor - piston advancement 2

Figure 26b. PLC inputs.

Module keys

No	Symbol	Function	Comment
B169	1	Z1 button	B.Start
B198	2	Z2 button	B.Stop
B183	3	Z3 button	B.Restart

Figure 26c. PLC buttons (B= button).

Physical outputs

Output	No	Symbol	Function	Comment	
Q1	B08	OUT	Discrete output	P1r	The piston 1 is withdrawn.
Q2	B09	OUT	Discrete output	P1a	The piston 1 is advanced.
Q3	B10	OUT	Discrete output	P2r	The piston 2 is withdrawn.
Q4	B11	OUT	Discrete output	P2a	The piston2 is advanced
Q5	B12	OUT	Discrete output	V	Suction cup acts.
Q6	B13	OUT	Discrete output	V*	Suction cup does not act.
Q7	B14	OUT	Discrete output	CRa	Rotating cylinder spins to the final position.
Q8	B15	OUT	Discrete output	CRr	Rotating cylinder spins to its original position.

Figure 26d. PLC outputs.

The Schneider PLC (series: SR2 B201BD is used for arm automation (figure 27).
Within the physical installation, the components which are framed in yellow outline are the optical sensors used in the assembly (figure 28).

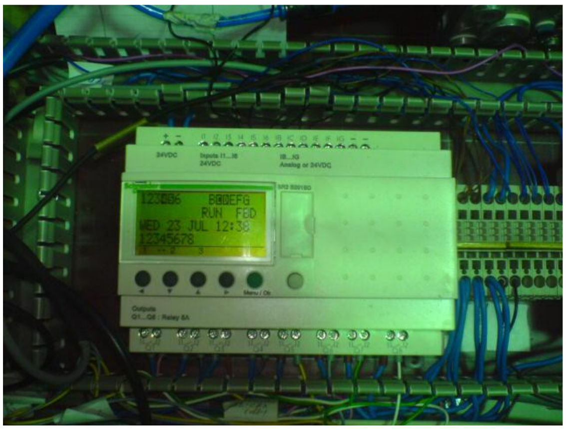

Figure 27. The Schneider PLC (series: SR2 B201BD used for arm automation.

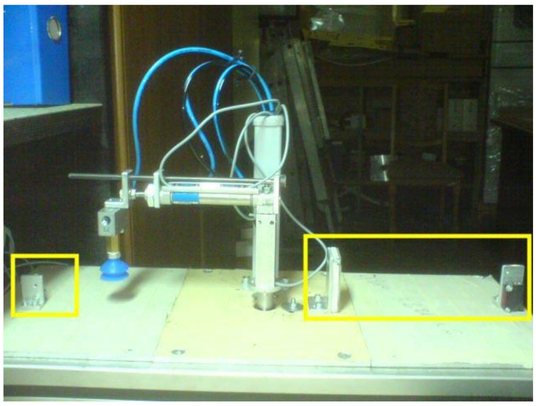

Figure 28. The physical installation; the components which are framed in yellow outline are the optical sensors used in the assembly.

5. Conclusions

Using Solid Works software package for robotic arm orientation in 3D virtual environment is very useful in industry because of the advantage of this automation to the industrial companies and its ability to change production quickly (almost without interrupting the flow / line in which is working the robotic arm). This handling system, is one of the most modern technologies currently used in the world, especially because of its versatility of use (in narrow spaces, for a variety of surfaces with partial or complete spins) and operational safety (applying high standards of ergonomic work rules).
Equipment features are the lifting process speed, the frequency of execution of manoeuvres and the manipulation process. The robotic arm can be easily implemented on a large scale in almost any industry where handling things. It has high accuracy and adjustable speed, so it can perform manipulation of objects as fast as you want. This robotic arm is equipped with two buttons besides the one to *start*, those are for *stop* and *restart*. The *stop* button helps if you want to stop the robotic arm in every stage, and the restart button helps you to bring it to its original position after you hit *stop* button, and when the arm is stocked. Thanks to these features listed above, it is not necessary the staff training and the robotic arm could be used by anyone.

6. References

[1] Lung-Wen Tsai 1999 *Robot Ananlist: The Mechanics of Serial and Parallel Manipulators he business model concept* (United States of America: John Wiley & Sons)
[2] Etienne Dombre 2007 *Modeling, Performance Analysis and Control of Robot Manipulators* (London: ISTE)
[3] Ivănescu A N, Tudorie R and Roşu A 2009 *Automate Programabile (Programmable machines)* (Bucharest: Politehnica Press)
[4] *Manual Automat programabil (Manual Automatic Programmable)* Zelio Schneider Electric 2007