Pneumatic cylinders controlled by two different controllers, Arduino and MyRIO: An educational approach

[1]Mihalache Ghinea, [1]Mihai Agud, [1]Mircea Bodog, [2]Mark Antonio Agud
[1]University POLITEHNICA of Bucharest, Faculty of Industrial Engineering and Robotics, Dept. of Robots and Manufacturing Systems, Romania
[2]University of Oradea, Faculty of Management and Technological Engineering, Romania

Received: June 21, 2021. Revised: January 13, 2022. Accepted: February 15, 2022. Published: March 14, 2022.

Abstract - Nowadays, new pneumatic equipment is becoming attractive to many industries that are beginning to accept the idea of replacing their hydraulic equipment with pneumatic one. Beginner engineers and students have a hard time understanding the differences between different controllers in the market and their applicability. This article provides information to support the understanding of the way the electronic control of stand-alone pneumatic systems works. This comparative study provides future specialists with the core knowledge concerning the influence that controllers have on the operation of pneumatic systems, as well as the principles of the controllers' utilization. The steps to take in controllers' utilization list the construction of the block diagram of an electronically controlled pneumatic system, followed by the simulation of the pneumatic system using modern software tools and ended by the assembly of the physical system and its programming by means of different classes controllers. If the control by pneumatic devices was still accepted, thus maintaining full-pneumatic systems on the market, the use of industrial electronic controllers would become indispensable both for more precise control of the systems and for increased industrial integrability. The compared results of a double-acting pneumatic cylinder control using the Arduino Uno and MyRIO-1900 controllers (academic equipment and for stand-alone applications) are presented. This way, the problem of pneumatic installations that requires equal forces to move the load in both directions is solved. The study performed a comparison of the times and cycles of the piston rod, variables that are defined in many applications. These results are also compared with those obtained by simulation using Automation Studio software (AS). The study was conducted to assess the interchangeability of both controllers in this common architecture. The research results are a step forward towards the implementation of electronic control in pneumatic systems using industrial controllers and then towards the harmonization of the structures thus established with systems in Industry 4.0.

Keywords: controller, IIoT, pneumatics, simulation

I. INTRODUCTION

Industry 4.0 (in Europe) or Smart Manufacturing (in the USA) are concepts that modern industry has already become accustomed to. Specialists quickly saw and accepted both the concept and especially its benefits [1,2]. Even if mechatronics dominates the conception and realization of modern industrial systems, Industry 4.0 through IIoT technology came with an advantage in the improvement and modernization of modern industrial processes [3, 4]. Thus, smart devices successfully emerged, and they consequently led to:

1. Production times up to 50% shorter;
2. The possibility offered to researchers to study and upgrade systems in real-time, benefiting from a data cloud;
3. Interfacing easily the systems with each other and industrial applications with each other, using Cloud with high-speed In / OUT access, Advanced Generation Cybersecurity and Big Data Analyzer (BDA);
4. The implementation of Predictive Maintenance in real-time helps the intervention of the technical teams to be done with great precision and speed without greatly influencing the continuity of the industrial process [5].

Industry 4.0 involves a lot of modern automation that mechatronics has already brought to a very high level in the past 20 years [6]. Industry 4.0 introduced a new concept: Industrial Internet of Things (IIoT), which allows almost all dynamic components to transmit through sensors and other real-time data acquisition components [1, 4].

Paradoxically, industrial systems are simple, but also more and more complex. In the technical academic environment nowadays, educational applications are being developed and they could be easily transferred to industry. Then the level of electronic control of that system should not be more complicated than it needs. The choice of controllers should be carefully made so as not to involve a very large budget or unjustified input of industrial intelligence (knowledge of automation, programming, and integration).

Choosing the controller for an automation application is sometimes a real challenge [7, 8]. First, it is important to know very well the application subject to automation, and the measure of a system's ability to increase or decrease in performance, the cost in response to changes in the application, and the system processing demands (scalability) [9]. The following should be considered: new or existing systems, discrete devices, environmental issues, loop control, analogue devices, speciality modules, I/O locations, communication, programming [10, 11].

Arduino and MyRIO are two of the most used types of controllers to solve an automation problem in a stand-alone system. They have many similarities but also differences.

The application proposed in the present article is a simple one, namely the control of a pneumatic cylinder with double action using the two controllers and highlighting some advantages and disadvantages of their use. There are also secondary objectives such as: knowing the Arduino and MyRIO controllers, how to program them, establishing the criteria for choosing one or the other according to the advantages and disadvantages offered.

A. *Current trends in the modern use of pneumatic systems*

When we refer to the command and control of pneumatic cylinders, there are two working possibilities: pneumatic and electronic command and control.

Pneumatic control is performed using pneumatic components while electronic control is obtained using specialized control equipment. The latter uses the signals received from the pneumatic equipment and based on a dedicated program, transmits commands that determine the operation of the system. It is worth mentioning the difference between electronic control and the electric actuation of some valves. In this situation, the control of pneumatic equipment is called electro-pneumatic [12, 13].

Our interest in this work is related to the use of electronic control systems (also called digital control) compared to pneumatic or electro-pneumatic control [14]. There are many differences between the two technologies, but one of the most important is related to the number of components controlled at the same time or at different times. The electro-pneumatic control consists of one element dedicated to a single pneumatic component, while the electronic one (such as PLC, PAC, etc.) is intended for several pneumatic components, even for an entire system [15, 16].

An important aspect in the electronic systems use is the choice of the controller, following the most important criteria mentioned above. Many specialists indicate very clearly the rules for choosing controllers, considering costs, the complexity of the controlled system, technical and mathematical-physical elements that dictate the functionality of the controlled system [7, 12].

In this article, we compare the use of two controllers on the market. One of them, Arduino, is very popular, having high integrability capabilities but also many restrictions regarding its use in complex systems [7, 8]. The second, MyRIO, is a product of the company National Instruments, very well adopted by the technical academic environment, having great potential for industrial applications [7, 8].

Considering the two controllers as being at the same level of integrability and acceptance by specialists, in our research we examined the ease with which one can control the operation of a common pneumatic component such as the pneumatic cylinder. Also, aspects of the programming of the two controllers and certain aspects related to their performance in the chosen application are analysed.

II. System Description

Pneumatic cylinders can be used in a wide range of applications. This is due to the wide variety of basic outputs, such as the force with which the load is displaced, the length of the stroke, the speed of movement of the load and sequencing (adding sensors can shorten the cycle times by eliminating time delays). Therefore, we did not focus on a particular application, but rather on controlling the operation of the cylinder using two of the best-known controllers. Previous studies explained very well the influence of load on the operation of the pneumatic system [12, 13, 17, 19]. On the other hand, a second reason why we were not concerned with the type and characteristics of the load was that the choice of a controller cannot be influenced by the type and characteristics of the load, but rather by the number of I / O, the integrability of the pneumatic application, budget, etc. [11, 13].

A. *Operational principle*

In figure 1 the block diagram of the studied pneumatic system is presented (the running is well highlighted also in the simulation scheme in AS - figure 4, in the physical assembly - figures 5 and 6 and in the electrical scheme – figure 7) [18, 21]. The electronic part is highlighted in red and the pneumatic part in blue. The simplicity of the structure is specifically chosen to reach precise and eloquent results; from this level on, the development is reachable in small and safe steps.

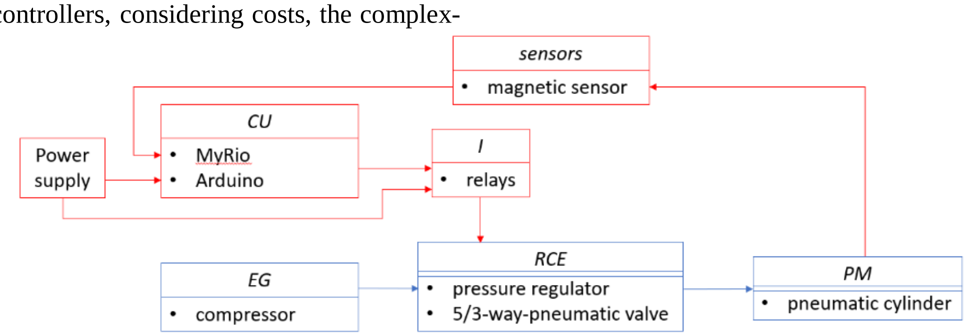


Figure 1. The block diagram of the pneumatic system

DOI: 10.46300/9109.2022.16.12 Volume 16, 2022

The model in Figure 1 is conceptual and does not form the basis of a mathematical study. Given its mathematical complexity, such a study can be described by partial differential equations. It is important to conduct an analysis using specific computing methods (especially parallel computing methods) in order to establish a program based on numerical methods to solve this type of engineering problem. The complexity of the mathematical model of the double-acting pneumatic cylinder has been addressed by many authors. In general, it is necessary to consider: the effects of nonlinear flow through valves, air leaks between rooms through constant or different sections, various delays or attenuations that occur over time.

Complexity is also given by a large number of control devices, sensors (force, pressure, flow), limiters, and tubes with different sections used in the system. Next, the block diagram in Figure 1 helps better understand the interaction between the various components of the system grouped into structural blocks.

The supply voltage, in this case, is transmitted to the controllers Arduino – 5V cc / MyRIO – 6-16V cc at CU (Command Unit) and to block I (Interface), materialized through the block of the relays (24V cc). The connection with the pneumatic cylinder (PM – Pneumatic Motor) is made on the one hand through the input element I that controls the RCE (Regulation and Control Element), materialized through the 5/3 valve, and on the other hand, through the magnetic sensors located at each end of the pneumatic cylinder. EG (Energy Generator) supplies compressed air to the RCE through an air preparation unit. RCE, commanded by I, sends the compressed air to the left/right chambers of the pneumatic cylinder, thus producing mechanical work, useful for moving the load.

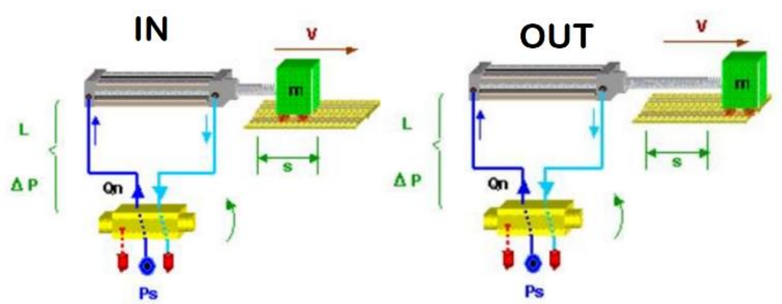
B. Purpose and Working

Today, due to its increasing performance, pneumatics tends to replace hydraulics, which in modern industry is considered too expensive and too polluting [2, 20]. The pneumatic cylinder or pneumatic motor is an essential component in pneumatics, and it is a mechanical device that converts the energy of compressed air/gas into linear motion carrying mechanical work in one or both directions of movement of the cylinder rod.

We set out to study the controllability of a pneumatic cylinder in the situation of using two common controllers on the market, considering at the same time, the results of the simulation of the respective process in a specialized application for pneumatics. Even if the industry requires complex automation, for integrated production/manufacturing lines, there are many stand-alone applications aiming to implement the benefits of automation at much lower costs than the systems belonging to Industry 4.0 [1, 2, 3].

In figure 2, we present the parameters of a pneumatic cylinder with double actuation, controlled by a directional valve: in the retracted position, IN (a) and the extended position, OUT (b). The directional valves are devices used to direct the flow of the fluid. These can be operated by a human operator, a pilot fluid, by an electrical signal or by a mechanical contactor (electromechanical systems) [21, 24].

If we mount two magnetic sensors on the cylinder and connect them in turn to a controller (Arduino and MyRIO, in our case), then the valve control can be done electronically, and the piston movement will be dependent on a software application, leaving aside the characteristics of pneumatic components and the gas used. The block diagram and the complete assembly of the pneumatic system are represented in figure 4 in AS, the software application used for simulation.



(a) (b)

Figure 2. The parametrization model of the double-action pneumatic cylinder: (a) retracted position; of the rod, (b) extended position of the rod.

Figure 3. Compressor used for compressed air supply

In figure 2: m – mass load (kg); Ps – supply pressure (4-10 bar); ΔP – pressure drop (0-2 bar); l – tube length (0.1-10 m); s – stroke (mm); Qn – normal flow (Nl/min); v – mass load speed (m/min). The study was performed for similar conditions of

loading (m), pressure (Ps), flow (Qn) and system architecture (length and diameter of tubes, type of valve, physical conditions of movement of the load, etc.).

C. Simulation of the pneumatic system with Automation Studio (AS)

Modern research methods of dynamic systems require the use of simulation and optimization software applications. In pneumatics, the Automation Studio application proved to be a useful and simple tool to study the dynamic behaviour of these systems [12, 22].

DOI: 10.46300/9109.2022.16.12 Volume 16, 2022

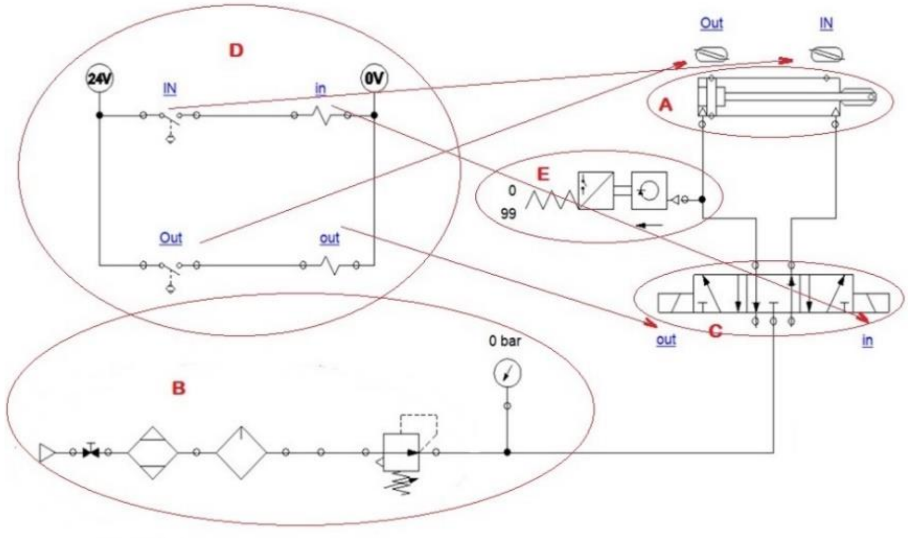
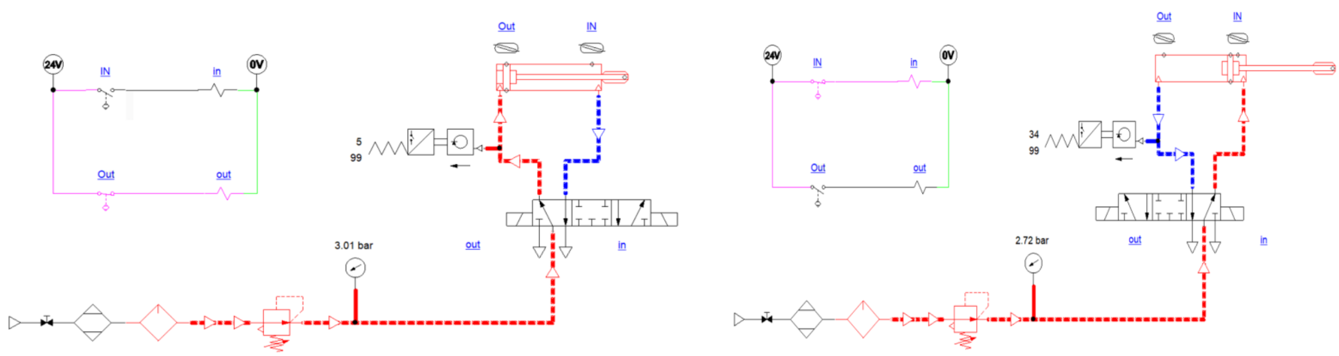


Figure 4. Diagram of the pneumatic system modelled in AS



(a) (b)

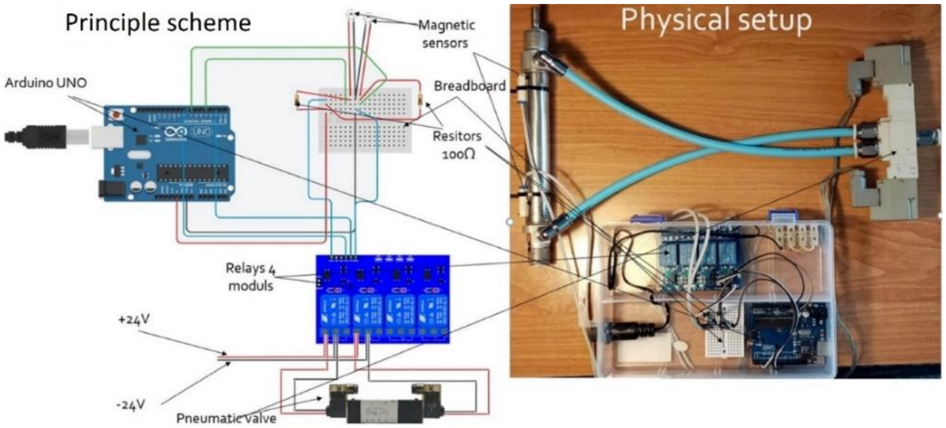
Figure 5. Diagram of the pneumatic system modelled in AS during the simulation: (a) the rod comes out - Out; (b) the rod comes in – IN

Figure 4 displays the model of the studied system, electronically controlled by a controller. The pneumatic cylinder with double action, A, receives compressed air from source B (through 5/3 valve C, which leads to IN/Out piston), this being directed through the valve 5/3, C. The components of block B are usually found in the physical systems in the air preparation unit (or FRL - Filter Regulator Lubricator). In AS, block D represents the electronic control through which we give the command to move the piston IN and Out by a signal sent to the valve coils 5/3. In AS, block E represents a counter that will count all the double strokes executed by the piston, useful in the statistical study of the analysis.

The two phases of the pneumatic system operation can be seen in figures 5 (a) and (b). In phase 1, when IN is open and OUT is closed, the left sensor detects the position of the piston (the presence of the magnet inside the cylinder, see figure 9, (a)), and Out moves the valve drawer to the right, allowing the air to pass through the left chamber (the red route) and to go out in the atmosphere through the room on the right after the blue route.

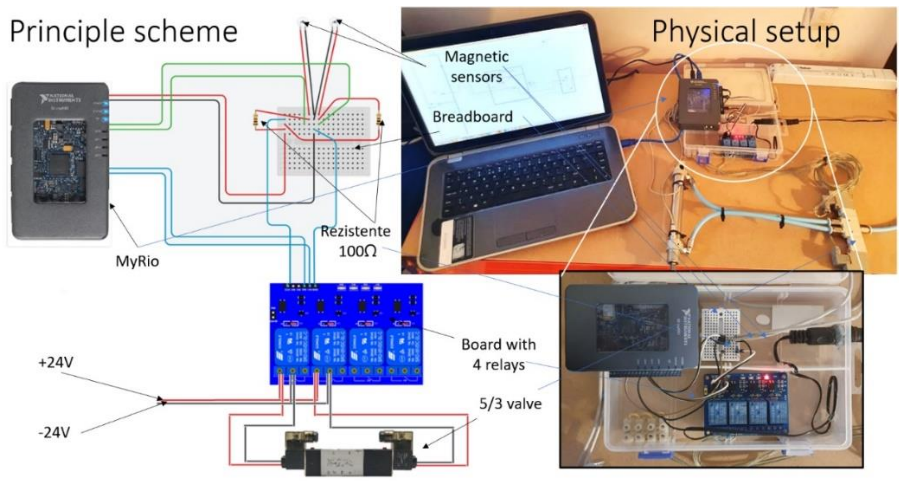
In phase 2, Out is open and IN closed, the sensor on the right detects the position of the output piston (the magnet on the right, see figure 9, (b)) and the "in" command of the valve drawer to move to the left. Thus, from the source, the compressed air enters the room on the right (the red route) and it is evacuated through the room on the left (the blue route). For special applications, the compressed air source can be, for instance, a tank with gas characteristics other than those of a continuous air compressed source.2.3. System configuration and details

In figures 6 (a) and (b) we presented the block diagram and the real assembly for the use of the Arduino Uno controller (Rev3), and in figures 7 (a) and (b) for the use of the MyRIO 1900 NI controller. As it can be seen, two similar assemblies were made, using the same type of sensors (Reed, Aventics ST6), the same block of 4 relays (powered at 24V, it uses a 5mA current to control, each relay being controlled separately by an optocoupler powered at 5V). Also, a double-acting pneumatic cylinder DSNU-20-200, with a piston diameter of 20 mm, was used as PM. The compressed air source was materialized by a STANLEY Fatmax 1.0HP, 6 Lt, 8 bar compressor (Figure 3). This compressor has a bottle with a volume of 6l, maximum suction air flow of 105 l / min, maximum pressure of 8 bar, maximum speed of 1450 rpm, low noise level of only 59 dB and no needs lubrication.



(a) (b)

Figure 6. The Arduino UNO utilisation: (a) The principle scheme; (b) The physical setup



(a) (b)

Figure 7. The MyRIO 1900 utilisation: (a) The principle scheme; (b) The physical setup

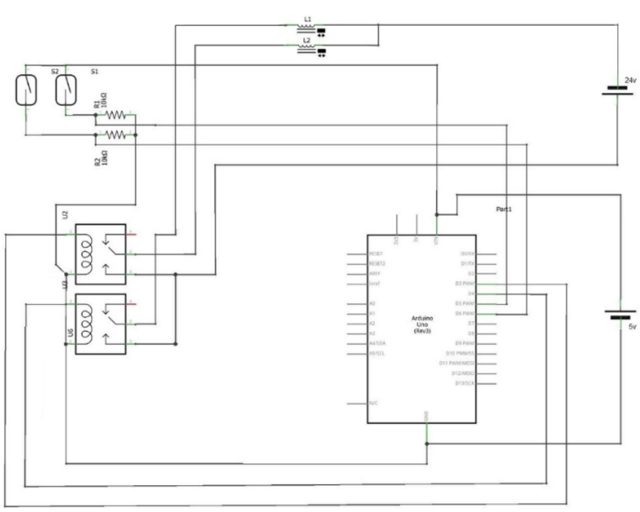
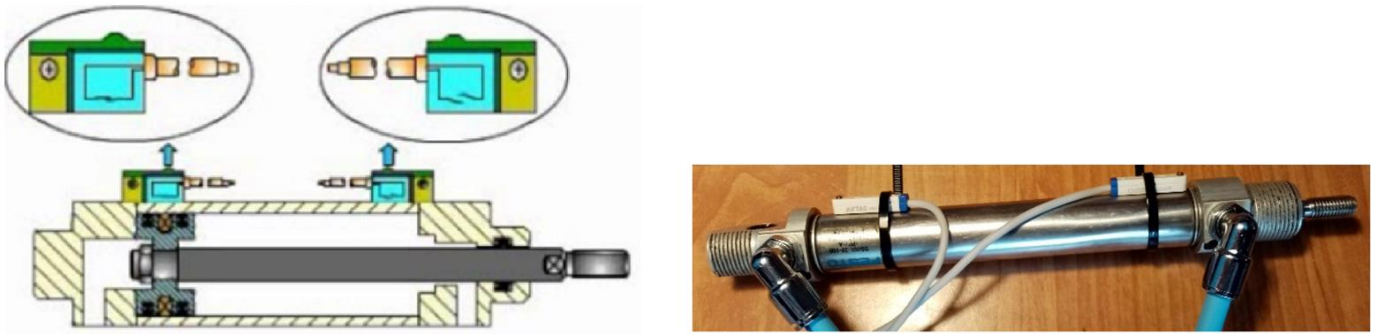


Figure 8. The electric scheme of the circuit

Figure 7 shows the electric diagram of the assembly, using Arduino (for MyRIO the supply voltage differs, 6-16V). The connection between the command/controller and pneumatic parts is made by the two magnetic sensors (the small white bricks with wires, Figure 9, b). The sensors are used to detect physical positions, pressures, flows, etc. The information collected by the operative part is sent to the control part through an input interface. The sensors can be divided into two sensor groups: digital (sometimes called "All-or-nothing") and analogue sensors. The digital sensors provide a Boolean signal that activates or deactivates the control element to which it is linked. The analogue sensors provide a real signal that is converted to a numerical value according to a pre-established scale.

The magnetic sensor reacts to the presence of objects disturbing the magnetic field emitted by the sensor. Placed against an aluminium cylinder, it reacts to the passage of the steel rod. The switch is then closed, and an electrical signal activates the component from the electrical control associated with this sensor [24].



(a) (b)

Figure 9. The pneumatic cylinder with double action: (a) principle scheme, AS design; (b) The physical cylinder with sensors mounted

III. Programming

The two chosen controllers can be programmed in totally different ways. For Arduino, an ADE / AIDE (Arduino Integrated Development Environment) is generally used but also other code editing environments [7]. When working with MyRIO 1900 from National Instruments we need to keep in mind that the working environment (not just programming) is LabVIEW [24], which is more similar to a "drawing program" than a Programming Language. LabVIEW offers a graphical programming approach that helps the user to visualize every aspect of the application, including hardware configuration, measurement data, and debugging. This visualization makes it simple to integrate measurement hardware from any vendor, to represent complex logic on the diagram, develop data analysis algorithms, and design custom engineering user interfaces [8, 23].

A. Arduino program

The source code in the IDE is written in C or C++. In section S1 (Figure 10) of the program, we define the variables and the pins to be used in the program. In the **void setup ()** in section S2 (Figure 10, block S2), the operation mode of the Input / Output pins is established, and the Serial Monitor is initialized. The **Serial Monitor** is an essential tool when creating projects with Arduino. It can be used as a debugging tool, to test out concepts or to communicate directly with the Arduino board [7, 25].

In the **void piston ()** function (Figure 10, block S3), the action of the piston is defined together with a timer divided in milliseconds which will display the total time of the action and the time in which a double stroke was performed. In the **void loop ()** function (Figure 9, block S4), the piston function, previously defined, is called, and the setting is established so that the number of double strokes performed by the piston is displayed every 10 seconds.

It is declared *sezIn1*, *sezIn2*, *relOut1 relOut2* on the pins which will be used, and the sensors, respectively the relays, are connected. Next, it is declared the variable *numar* that counts the Double Strokes (DS) that the piston makes. This number must be set to 0 at the beginning of the program. It is declared the variable *isNew* that will help to reset the calculation for each DS, it is declared the variable *period*, which adjust the time range the races are counted, set by us at 10000 milliseconds, ie 10 seconds. It is declared the variable *time_now* using a declaration form for more memory, which is used to calculate the time interval for displaying the number of DS in the set time. It is declared the variable *prevNumber* which must be equal to 0 at the beginning of the program. This variable helps to calculate the number of DS in the established time interval. For more memory, in the same way, it is declared the variable *previousMillis*, which must be equal to 0. This variable will help to calculate the time between the 2 races.

S1

```
int senzIn1 = 4;
int senzIn2 = 7;
int relOut1 = 2;
int relOut2 = 8;
 int numar = 0;
 int isNew=0;
 int period = 10000;
 unsigned long time_now = 0;
 int prevNumar=0;

 unsigned long previousMillis = 0;
```

S2

```
void setup() {
  Serial.begin (9600);
  pinMode( senzIn1, INPUT); // s1
  pinMode( senzIn2, INPUT); // s2
  pinMode( relOut1, OUTPUT); // b1
  pinMode( relOut2, OUTPUT); // b2// put your setup code here, to run once:
```

S3

```
void piston()
{
  if (digitalRead(senzIn1) == HIGH and digitalRead(senzIn2) == LOW )
  {
    digitalWrite (relOut2, HIGH);
    digitalWrite (relOut1, LOW);
    if(isNew==0) {
      unsigned long currentMillis = millis();
      unsigned long iterationMillis = currentMillis-previousMillis;
      previousMillis = currentMillis;
      numar++;
      Serial.print("[");
      Serial.print(currentMillis);
      Serial.print("]");
      Serial.print("\t");
      Serial.print(numar);
      Serial.print("\t");
      Serial.print(iterationMillis);
      Serial.println("");
      isNew=1;
    }
  }

  if (digitalRead(senzIn1) == LOW and digitalRead(senzIn2)  == HIGH)
  {
    digitalWrite (relOut2, LOW);
    digitalWrite (relOut1, HIGH);
    isNew=0;
  }
}
```

S4

```
void loop() {

  piston();
  if(millis() >= time_now + period){
      time_now += period;
      Serial.print("Ten seconds iterations: ");
      int deltaNr=numar-prevNumar;
      Serial.println(deltaNr);
      prevNumar=numar;
   }

}
```

Figure 10. The program for Arduino divided into 4 sections

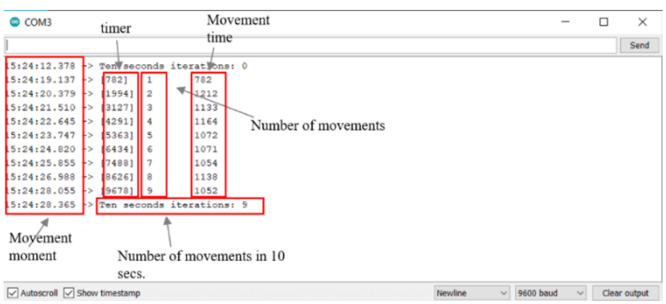


Figure 11. The screenshot with the results of the Arduino program running

Figure 11 is a screenshot displaying a sequence of running the program with the data indicated in the description above, the *number of movements* being the number of double strokes (DS) of the IN / OUT piston. Figure 15 graphically shows the cylinder dynamic for the two controllers, respectively the travel time for each DS for 10 seconds. There are some very useful observations presented in the next chapter, influenced not only by the capabilities of each controller but also by the characteristics of certain components of the pneumatic system. Note that the application LabVIEW can be also used as a graphical interface for the Arduino controller [7, 25].

B. MyRIO program

The world of modern technology (IT&C, robotics, mechatronics, computing machines, Smart technologies, etc.) is not only constantly changing, but also evolving at a very high speed. While conducting scientific research on certain devices of a certain generation, some new ones are already ready to be launched on the market.

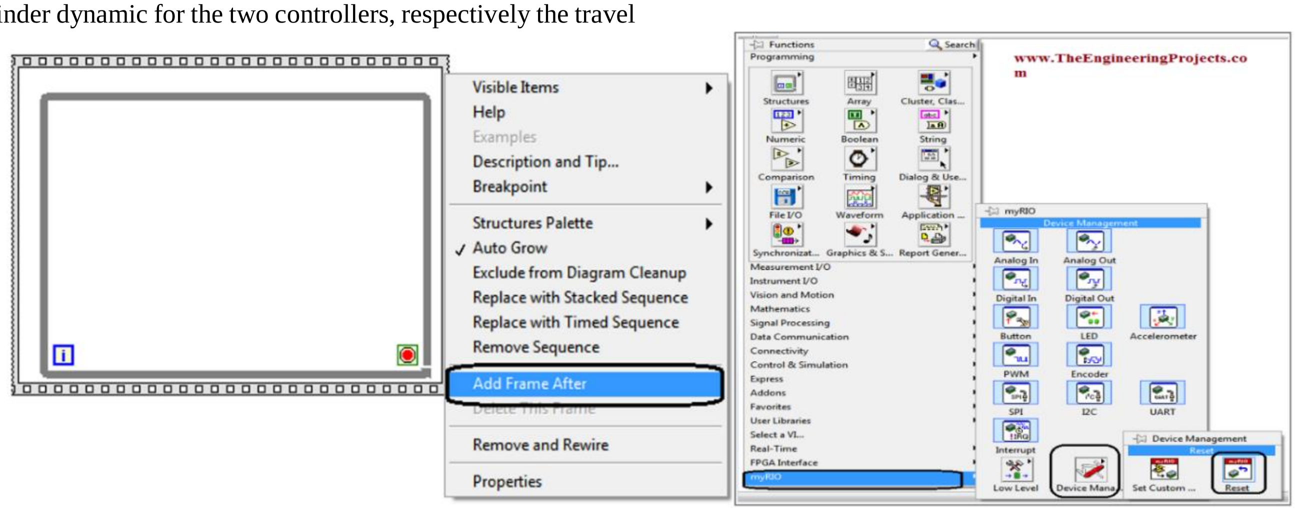


Figure 12. LabVIEW working environment

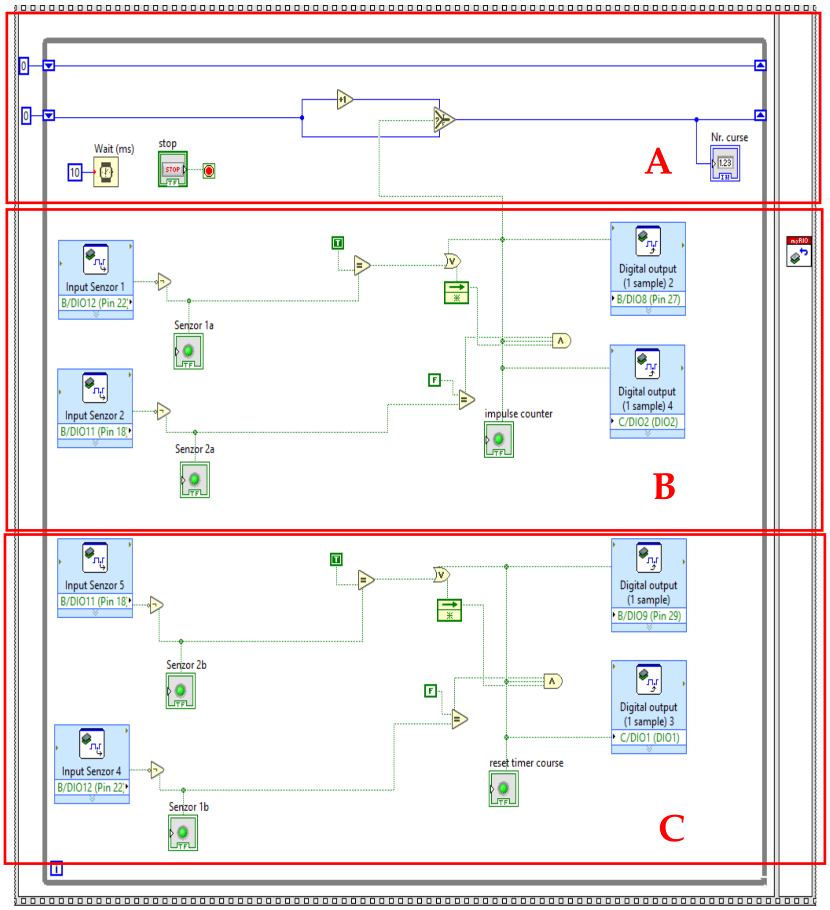


Figure 13. Block level implementation in LabVIEW

Even so, we chose to work with the MyRIO 1900 version of 2019. [26]. LabVIEW programming means using the Graphical Programming Language (GPL), a programming development environment that puts C, C ++ or Java code behind the graphical elements of object-oriented programming (OOP). LabVIEW uses a graphical programming language, often called "**G**," to create programs in a pictorial form called a *block diagram* [26].

Figure 12 exemplifies the pictorial way of working in LabVIEW, through which the graphic code is created with the help of graphic elements. The Virtual Tool (IV) in LabVIEW contains three modules: **Front Panel** (an interactive user interface), **Block Diagram** (an executable program that after establishing the program diagram verifies and validates its functionality),

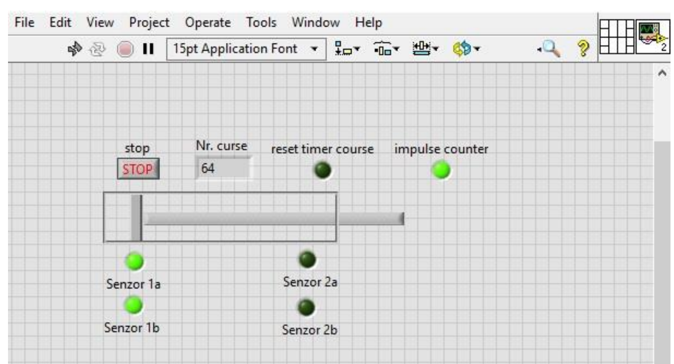


Figure 14. Front Panel

and **Icon** (allows the creation of subroutines inside a Block Diagram).

The written code in figure 13 allows the piston to be operated according to the reading of the signals from the sensors and calculates the time required for a piston stroke (DS). Block A counts the number of DS, the total time, and the time between two DS. Block B moves the piston to the position of sensor 1 and block C moves the piston to the position of sensor 2. Figure 14 shows the Front Panel, which is simple and intuitive and contains "LEDs" that identify the two positions of the two sensors, the DS number, and a general STOP button.

IV. Results and Discussion

The differences between the simulation of the electronically controlled pneumatic system and the study on the experimental bench using the two controllers dedicated to Stand-Alone systems are important and worth remembering for researchers. The mathematical apparatus used by the AS application was implemented without considering one controller or another on the market. The designers and manufacturers of the product used the mathematical relations established in the theories of physics, mechanics, and fluid mechanics [22].

Figure 15 graphically presents the important variables of the pneumatic system in the simulation with AS. The X-axis shows the time in seconds, and the y-axis shows the pressure [bar], the stroke length [mm] and the linear velocity [cm/s]. Figure 16 is

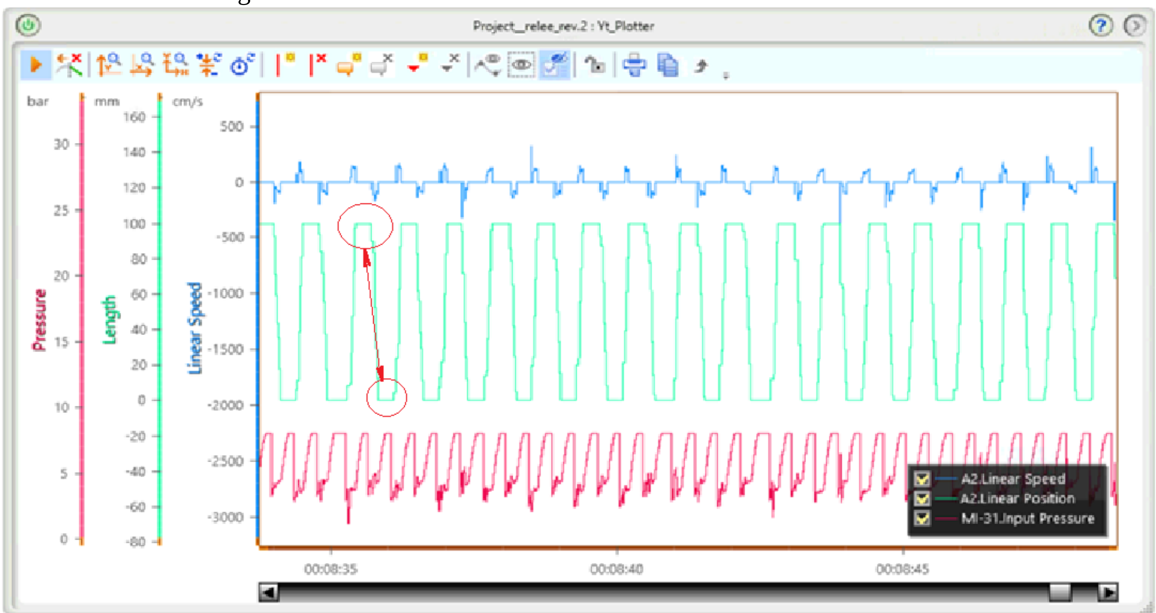


Figure 15. The representation of the important variables of the pneumatic system in the simulation with AS

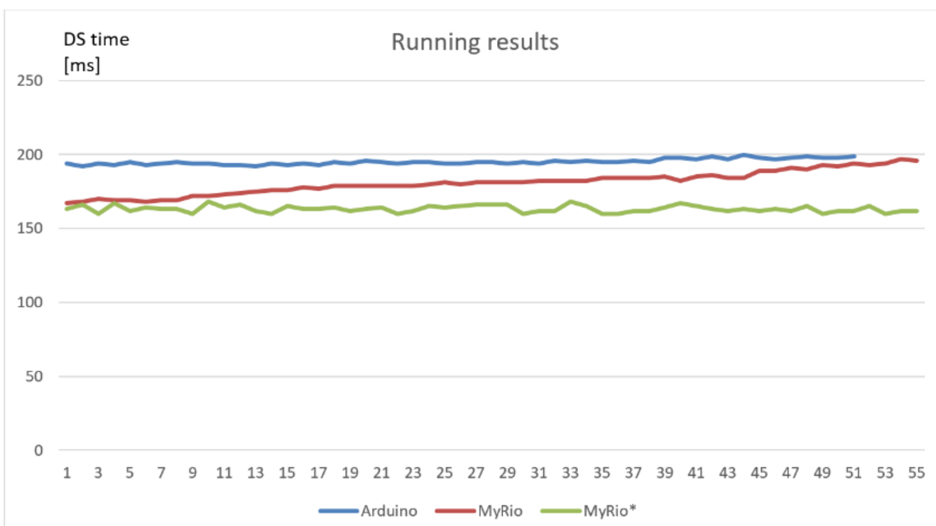


Figure 16. The slope of the DS variable in the Arduino and MyRIO cases

the graphical representation for 10 seconds (with the number of CDs on the x-axis, and the duration of a CD on the y-axis).

A first observation is related to the number of double strokes DS made by the piston in the 10-second set as the reference interval. In the simulation with AS, DS is much smaller (13 CD) than the number of DSs in the real analysis. Even here, there is a difference between the situations using the two controllers. Using Arduino, 51 DSs were obtained whereas using MyRIO, 55 DSs were obtained. The same pneumatic equipment was used in both assemblies, so the parameter changes were identical during the measurements.

For a pressure of 6.3 bar with a load of 0 kg, the maximum speed that the piston can reach is ~ 1.2 m / s. The results indicate that the piston reached a speed of ~ 0.55 m / s, ie about 46% of the maximum speed. The limits of the pneumatic system were therefore not reached even by half, the difference up to the theoretical maximum limit being due to the difference between the testing pressure and the pressure used in the theoretical calculations, but also due to losses within the system. The frequency of the processor in the Arduino board is 16/48 MHz and the frequency of the MyRio 1900 controller is 667MHz. These two values are much higher than the frequency of the pneumatic system, so can be used both of these controllers without influencing the dynamic behaviour of the pneumatic system. The difference between the two controllers, when it is necessary to choose between them, is given by the difficulty of use, the price, and the complexity of the pneumatic system which must be controlled. Arduino has the advantage of simplicity but is limited by the complexity of tasks, and by the internal memory. MyRio is used in academia, requires a more advanced level of training but can be implemented within complex systems, offering many operations and a professional user interface programmed in LabView. These advantages make MyRio usable in research but also in non-industrial technical applications.

Returning to the diagram in figure 15, it is easy to observe the two stationary pistons at the two ends of the cylinder (the circled areas in the figure), which represents stagnation due to the software running [23]. In the diagram in figure 15, there are three curves: two for the case when the MyRIO controller was used, and one for the Arduino's. For pneumatic systems that require smoother and more constant movement, the use of Arduino seems preferable, even though the average duration of DS is still constant in the unit of time with smaller deviations than in the case of MyRIO. The same constant could be obtained by using MyRIO; however, for certain corrections, a compressor with a larger tank should be used. This would ensure a constant flow for a longer period. Also, the pneumatic architecture of the system should be modified, either with additional devices or with others with greater adjustment possibilities.

The difference between the MyRIO and MyRIO * curves is explained by the variation of the working pressure in the installation. In the case of the MyRIO curve, a compressor with a small capacity of 6 litres was used, while in the case of the MyRIO * curve, a constant pressure source was used. For users, it is good to remember the influence of the pressure in the pneumatic system on the performance of the controllers, given that this is a situation that would certainly be repeated if industrial controllers were used (PAC, PLC, SCADA, etc.).

V. Conclusions and Future Scope

MyRIO 1900 runs a real-time operating system based on Linux; it can run a LabVIEW code on it, having utilities to allow configuration/use and also multithreading to have some parallel operations on MyRIO. Doing this on Arduino is much more difficult, given that there is no multithreading/use of interrupts. The FPGA on MyRIO allows to run, acquire data, filter, or very high-speed data I / O at its clock rate, which is much faster than an Arduino could work. MyRIO is more powerful, has a faster processor and more memory etc. MyRIO has several I / O options - analogue / digital IO, RS232 / SPI / I2C etc. The data acquisition system (DAQ) for analogue inputs/outputs on MyRIO is better / faster than Arduino, but Arduino is 10 times cheaper. Using LabVIEW, debugging code is easier than with Arduino. Arduino is troubleshootable by trial and errors evaluation, while with LabVIEW and its interactive mode, it can be tested and debugged more easily.

Industrial automation is crucial in modern production. Unfortunately, the complexity of the new production systems integrated through Internet-IIoT (Industry 4.0 or Smart Manufacturing) grows much faster than the training improvement of human resources in the field. For this reason, in technical schools and universities the use of controllers in Stand-Alone systems, in our case for pneumatics, becomes a mandatory step. In any industrial application, the behaviour of the pneumatic system is very little dependent on the level of integration, and on the behaviour of other equipment and systems. Under these conditions, the architecture presented in this article is sufficient to study the behaviour of a pneumatic cylinder when it is controlled by various controllers. It was highlighted that the simulation of the pneumatic system working with electronic control has certain deviations from the real measurements, which must be considered by specialists who implement such systems.

Compared to previous research in which a fully pneumatic motor was used, and the control was performed by pneumatic equipment, the present research highlights important differences [12]. Using controllers means increased flexibility and integrability. Shortening the pneumatic chain by introducing controllers brings financial benefits, even though the design-implementation team must also include a specialized programmer. Future research includes the use of industrial PAC or PLC controllers, and then the integration through IIoT of electronically controlled pneumatic systems in applications for Industry 4.0. Generally, certain controllers used in pneumatic systems are treated individually in the literature, while their comparative analysis is rather scarce. This paper displays the specialists the whole spectrum of the implementation of chosen controllers within the same pneumatic system, starting from operation simulation, using dedicated software, to the specific programming and data analysis.

References

[1] Tay, S. I., Chuan, L. T., Nor Aziati A. H., Ahmad, A. N. A. "An Overview of Industry 4.0: Definition, Components, and Government Initiatives", Journal of Advanced Research in Dynamical and Control Systems, vol. 10, no. 14 - Special Issue, 1379-1387, 2018.

[2] Petrillo, A., De Felice, F., Cioffi, R., Zomparelli, F. "Fourth Industrial Revolution: Current Practices, Challenges, and Opportunities", In Digital Transformation in Smart Manufacturing, Petrillo, A., Cioffi, R., De Felice, F., IntechOpen, London, UK, pp.1-20, 2018.

[3] Munirathinam, S. "Industry 4.0: Industrial Internet of Things (IIoT)", In Advances in Computers, Raj, P., Evangeline, P., Eds., Academic Press, Cambridge, MA, USA, vol. 117, pp. 129-164, 2020.

[4] Lampropoulos, G., Siakas, K., Anastasiadis, T. "Internet of Things in the Context of Industry 4.0: An Overview", International Journal of Entrepreneurial Knowledge, vol. 7, pp. 4-19, 2019.
[5] Tran Anh, D., Dabrowsky, K., Skrzypek, K. "The Predictive Maintenance Concept in the Maintenance Department of the "Industry 4.0" Production Enterprise", Foundations of Management, vol. 10, pp. 283-292, 2018.
[6] Vieira, A.A.C., Dias, L.M.S., Santos, M.Y., Pereira, G.A.B., Oliveira, J.A. "Setting an Industry 4.0 Research and Development Agenda for Simulation – a Literature Review", International Journal of Simulation Modelling, vol. 17, pp. 377-390, 2018.
[7] Sarmad, S., Salloom, M. Y. "Arduino-Based Controller for Sequence Development of Automated Manufacturing System", Al-Khwarizmi Engineering Journal, vol. 16, pp. 22-33, 2020.
[8] Makowski, T., Domin, J. "cRIO-9022 and myRIO-1900 controller output state change rates", Szybkobiene Pojazdy Gsienicowe, vol. 50, pp. 139-151, 2018.
[9] Krokavec, D., Filasova, A., "Control Design for Linear Strictly Metzler Time-delay Systems", WSAS Transactions on Systems and Control, vol. 16, pp. 519-526, 2021.
[10] Nguyen, H. M. "Nonlinear Feedback Linearization Control for Wind Generators in Hybrid Wind Energy Conversion Systems", WSAS Transactions on Systems and Control, vol. 16, pp. 493-501, 2021.
[11] Madhanamohan, K., Praveen, R.K., Nirmalraja, T.R., Goutham, H., Sabarinathan, R., Logeesan, A. "Industrial Automation System", Advance in Electronic and Electric Engineering, vol. 3, pp. 717-726, 2013.
[12] Ghinea, M., Agud, M., Bodog, M. "Simulation of Pneumatic Systems Using Automation Studio Software Platform", International Journal of Simulation Modelling, vo. 19, pp. 655-666, 2020.
[13] Santos, A.A., Ferreira da Silva, A. "Electro Pneumatic Control versus Grafcet", Proceedings of International Congress on Education, Innovation and Learning Technologies, Granada, Spain, 2015.
[14] Miyajima, T., Fujita, T., Sakaki, K. "Development of a digital control systems for high-performance pneumatic servo valve", Precision Engineering, vol. 31, pp. 156-161, 2007.
[15] Lewis, R.W. "Programming industrial control systems using IEC 1131-3", Revised edition, The Institution of Electrical Engineers, London, UK, pp.1-28, 1998.
[16] Arnold, Z. "What's the Difference Between a PLC and PAC?", Machine Design, https://cdn.machinedesign.com/files/base/ebm/machinedesign/document/2019/04/machinedesign_11117_wtd_plcpac_pdflayout.pdf, (accessed on 12 July 2021).
[17] Kheiralla, A.F. "Design and Development of a Pneumatic Circuits Bench for Education Purposes", Proceedings of International Conference on Engineering Education, Coimbra, Portugal, 2007.
[18] Gajsek, B., Marolt, J., Rupnik, B., Lerher, T., Sternad, M. "Using Maturity Model and Discrete-Event Simulation for Industry 4.0 Implementation, International Journal of Simulation Modelling, vol. 18, pp. 488-499, 2019.
[19] Foit, K., Banas, W., Cwikla, G. "The pneumatic and electropneumatic systems in the context of 4th industrial revolution", IOP Conference Series: Materials Science and Engineering, vol. 400, pp. 1-6, 2018.
[20] Perera, F. "Pollution from fossil-fuel combustion is the leading environmental threat to global pediatric health and equity: solutions exist", International Journal of Environmental Research and Public Health, vol. 15, pp.1-17, 2018.
[21] Figliolini, G., Rea, P. "Design and Test of Pneumatic Systems for Production Automation", Proceedings of the Canadian Design Engineering Network Conference, McGill University, Montreal, Canada, 2004.
[22] Ojstersek, R., Acko, B., Buchmeister, B. "Simulation Study of a Flexible Manufacturing System Regarding Sustainability", International Journal of Simulation Modelling, vol. 19, pp. 65-76, 2020.
[23] Famic Technologies. "Automation Studio Educational: Getting Started", from: https://www.famictech.com/en/Support/Automation-Studio/Getting-Started#quickstartguides, (accessed on 20 June 2021).
[24] Jonnala, P., Satrasupalli, S. "Semi-Automated Polyhouse Cultivation Using LabVIEW", International Journal of Computer Science and Business Informatics, vol. 7, pp. 1-13, 2013.
[25] Rajendran, V.K., Chockalingam, A. "Development of Design of Pneumatic Circuits using Arduino Microcontroller", International Journal of Science Technology & Engineering, vol. 5, pp. 27-29, 2018.
[26] Travis, J., Kring, J., "Introduction to Graphical Programming with LabVIEW", In:LabVIEW for Everyone: Graphical Programming Made Easy and Fun, 3rd Edition, Prentice Hall, Hoboken, USA, pp. 3-21, 2006.