Software Development for Field Level Industrial Process Control Network System

Oleksandr Dobrzhanskyi[a], Martin Bogdanovskyi[a], Andrii Tkachuk[a] and Tetiana A. Vakaliuk[a,b,c]

[a] *Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine*
[b] *Institute for Digitalization of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*
[c] *Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine*

Abstract

The results of the presented researches dial with the software design study methods relatively to the industrial process control network-system on field level. Configuration and programming stages for base elements are investigated and substantiated. Students obtaining a bachelor's or master's degree do not have the opportunity to acquire knowledge in a structured manner precisely on the issue of assembling distributed automatic systems. It would be strange if it weren't true, but the knowledge of networked distributed automation systems still needs structuring and generalization. We developed a methodology concept to structure and generalize knowledge regarding the construction of distributed network automated control systems based on an industrial field network for information exchange. Only a systematic approach in the organization of education in the field of distributed network automation systems can guarantee quality education for future engineers. This article presents an attempt to structure the necessary educational material. Representation of the ultimate learning goal in the form of a universal networked automation system that has the ability to be distributed thanks to industrial field network protocols. In the article, we analyzed a variety of material on certain aspects of the development of a distributed network automation system, systematized this material and summarized it, thereby forming the structure of educational content for textbooks on industrial automation.

Keywords [1]

Distributed network, automated control systems, the assembling of automatic system

1. Introduction

The practical implementation of control systems based on modern approaches makes it a problem for developers or system integrators to deal with choosing the software and hardware architecture of the future system. Thus, during the training of specialists in automation and computer-integrated technologies, the presentation of the most important stages of synthesis of a process control system that uses modern programming languages, special methods of debugging, network management, involvement of configuration characteristics of individual system elements are required. As already mentioned above, today there are no concrete educational and methodical materials or textbooks that would summarize the process of creating distributed network automation systems. There are only separate user instructions for a limited group of narrow specialists. Whereas the educational process requires a systematic approach to solving the issues of structuring educational material. Instrumental structuring is present in production conditions, but this has not yet been reflected in the educational process. Creating a final educational product is relevant today: textbooks, methodological guidelines, curricula, programs of laboratory workshops or industrial practices, which is not possible without

CITI'2023: 1st International Workshop on Computer Information Technologies in Industry 4.0, June 14–16, 2023, Ternopil, Ukraine
EMAIL: aikt_doo@ztu.edu.ua (O. Dobrzhanskyi); martin.bogdanovskiy@gmail.com (M. Bogdanovskyi); andru_tkachuk@ukr.net (A. Tkachuk); tetianavakaliuk@gmail.com (T. A. Vakaliuk)
ORCID: 0000-0002-4330-0287 (O. Dobrzhanskyi); 0000-0003-2870-4248 (M. Bogdanovskyi); 0000-0003-2466-6299 (A. Tkachuk); 0000-0001-6825-4697 (T. A. Vakaliuk)

researching the set of issues that need to be presented to the applicant. The developer of the educational product should be guided by the recommendations developed in this article, and the acquirer of the educational product should receive a high-quality modern level of education, in accordance with the requirements of the labor market and the needs of Industry 4.0.

2. Literature review

Today, the leading manufacturers of industry equipment are focusing on expanding the range of universal and interchangeable element base of automatic devices as a response to customer needs in control and management usability [1], [2], [3]. In paper [1] it is indicated that the research and development in the field of automation includes programmable logic control (PLC), robotics, distributed control system (DCS), computerized numerical control machine (CNC), radio frequency identification (RFID). The design of intelligent systems for scheduling and manufacturing the product such as flexible manufacturing systems (FMS), computer aided manufacturing (CAM), computer integrated manufacturing (CIM), lean manufacturing and green manufacturing are considered as well. Paper [2] discusses the practical use context of HMIs in smart factories and offers HMI recommendations for users, designers, and researchers. In work [3], a device named IoT-PLC is designed and prototyped, in an effort to generate a PLC tailored for the Industry 4.0 revolution. The proposed IoT-PLC operates as a containerized piece of equipment, with each functionality packaged within a separate container. It uses a virtual device model that works as an abstraction method to represent real entities, so that IoT-PLC applications can interact transparently and with straightforward compatibility with upper cloud layers.

The main attention today is focused on the hardware compatibility of particular elements of automation systems. The manuscripts [4-5] address improvements for PLC network availability by using correct data slicing at the application level along a tuned transmission rate in accordance with the noise levels of the power grid. Successful communications, even at low rates, mean that no manual interaction from energy supplier operators is needed to reduce the maintenance costs for both the energy companies as well as for the end user.

On papers [6], [7] are describing the so-called enabling technologies and systems over the manufacturing environment. Providing the possibility of creating many hardware structures, manufacturers particularly avoid the question of creating methods for synthesizing the software of automated control systems, their full composition, open for further research.

The results of recent applied and theoretical research [8] in the development of methods for the synthesis of distributed automated control systems contain mostly classification of future expected properties, network architecture, remote control, a list of modern communication technologies and software components that can be used for synthesis.

Previously, the field of automation and management had to be provided only with terminological support, regulations of forms and order of document circulation. It is typical that the stages of automated control systems creation are still supported, however, only by the organizational and general tips of the engineers of automation implementation. Thus, the issue of developing educational methods for the synthesis of software for automated control systems with a modern component base has not been studied.

Without a systematic approach to this problem, sustainable development of the engineering education system [9] is impossible. The relevance of the chosen topic is also confirmed by the opinion of experts, who believe that the lack of necessary elements in the education of a future engineer critically affects social development in general, including the development of industrial capabilities of modern society [10], [11]. Moreover, the lack of attention to at least one educational component leads not to the stagnation of the process of industrial development, but even to the degradation of industry and a decline in the economy. Finally, there is a consensus that engineering education needs to be developed to support sustainable development [12], [13].

3. Methodology

The only methodological approach in finding ways to improve learning in the field of distributed network systems of automatic control is, in our opinion, the way of analysis, generalization and construction of the structure. This is supported by a cohort of specialists in the educational field that prepares future automation engineers [14], [15], [16], and distributed network systems also researched in [17], [18], [19], [20].

Regarding the analysis, the following actions were performed: we analyzed the results of specialized google-search-request, such as "(study OR learn OR education) materials (automatic control OR automatic system) AND industrial networks site:*.edu", especially concerning educational resources. It was revealed that there is almost no engineering educational literature on the methods of building automatic control systems based on industrial networks in such a way that the knowledge obtained in this way could be used precisely in engineering practice in industry, or in the practice of educational projects.

Regarding the generalization. The structural elements that should be in the educational content are highlighted. First of all, attention was paid to components that are missing or that are considered inaccurately. This mainly concerns the elements of learning to create visualization, network connection, interaction in a multi-component industrial network, creation of concrete software modules with network capabilities, the unity of the entire network system as a single workspace, including the capabilities of interaction with the equipment operator.

Regarding the structuring. Here there is a need to repeatedly revise the order of submission of educational material. Finally, it is necessary to try to transfer production experience to elements of training. Introduce an element of development into the learning process. There is a need to emphasize the universality of the educational content, with the help of which a future engineering student could design a certain variety of automation systems with network connections.

4. Results

4.1. Proposed work

Before initializing the educational process or any educational course around the field level of modern distributed control systems it is necessary to plan the main stages, to take in mind what device-structure could be involved in laboratory sets. And, as a result, the strict requirement is to evaluate hardware and software obstacles that can appear in laboratory activity. All of these are far described.

Defining hardware requirements

The content of the software debugging process of the distributed control system is determined by the architecture composition model. A typical set of modern automation equipment designed to control technological processes can be presented into the following basic components (see figure 1):

1. techniques of measuring the parameters of the technological process according to various sensors with analog, discrete or digital output;
2. techniques of storage and execution of control programs according to programmable logic controllers (PLC);
3. techniques of collecting and processing of raw data to technological parameters according to modules of input/output analog and discrete signals;
4. tools that provide human-machine interface (HMI) according to graphic and text panels of the operator;
5. techniques of communication according to industrial field buses, local and global Internet;
6. techniques of setting the configuration and creating the software part of the automated process control system according to programming environment of controllers, visualization, SCADA-system, various software configurators, OPC and WEB servers;
7. special purpose power equipment, which presents as an object to control or is designed to provide certain modes of actuators and drives operation. There are power devices controlled by standard levels of analog and discrete control signals: drives, power variators and starters.

The experience of system synthesis on the basis of the presented components shows that the most efficient design in productivity means and control systems fast development will be achieved in case of use of components group 2-5 from one manufacturer. This fact is connected with the still particularly unresolved issue of absolutely guaranteeing the compatibility among automated control system elements of different manufacturers according to the standards of industrial field buses.

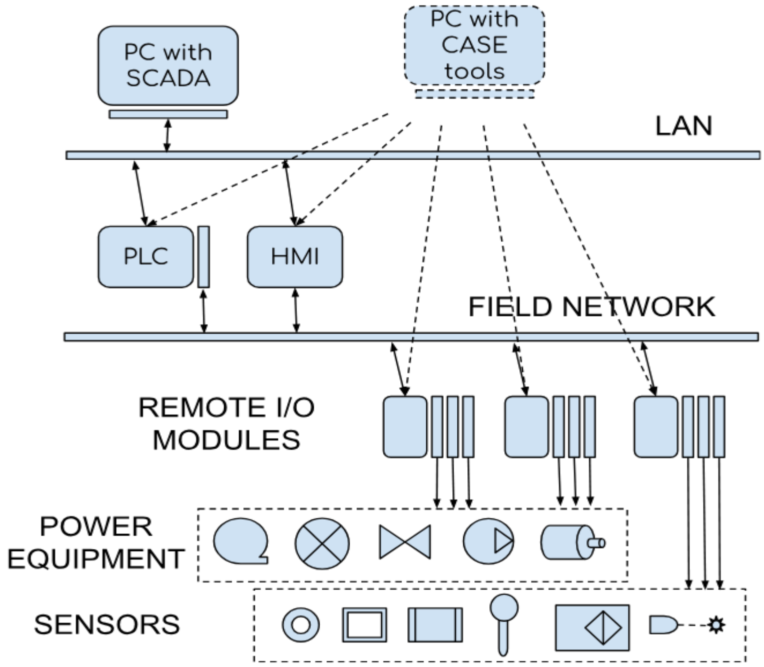

Figure 1: The common structure of the automated control network system

Definition of software requirements

Modern industrial automation concept definitely needs software support. Software systems must be developed as tools such as computer aided software engineering (CASE) and must support the generalized international standards IEC 61131-3, which allows their systematic study and use as an integral part of the theoretical representation of the automated control systems synthesis. The CASE provides a set of software development tools with elements of computer support such as templates, standard blocks, standard functions, automatic code verification, code and program interface blanks. IEC 61131-3 is an industry standard that defines five programming languages for PLCs.

Definition of requirements for stages of automated control system synthesis

In case of modern equipment use, the creation of a process control system involves the following initial stages (figure 2):

1. the stage of direct design of the system (the equipment selection and combine at the physical level);
2. the stage of setting the configuration of the interfaces of individual devices, in order to further connect them at the software level;
3. the stage of programming of the control devices of the automated control system.

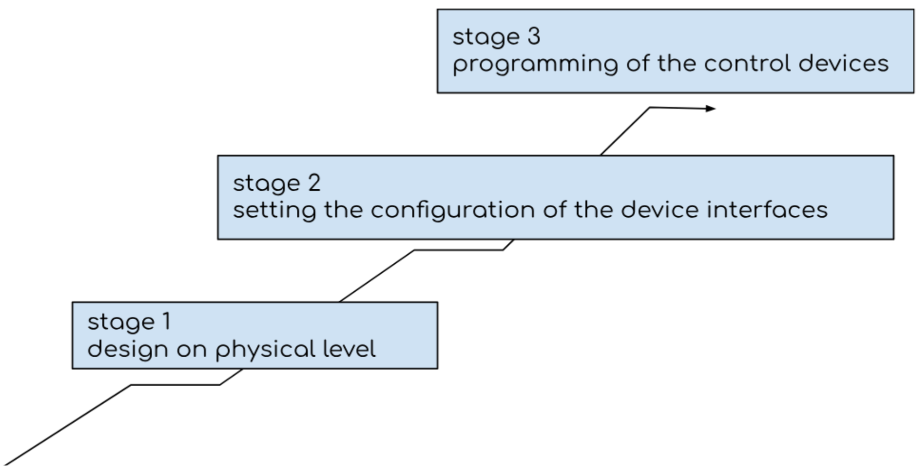

Figure 2: The common stages of the automated control network system synthesis

Definition of the content of the interface configuration and programming stages

At the stage of programming and configuration, the system is already mounted and the devices are already connected to the network lines. However, in time of accessing devices from a computer to set certain device parameters, a conflict may occur because the network configuration of the devices has not yet been set and the devices may have the same network addresses. Which may cause a network conflict. Therefore, to avoid this case would be rather to turn off the power of one of the conflicting devices, or disconnect all conflicting devices from the network and, connecting them in certain order, set the configuration settings or download the control program.

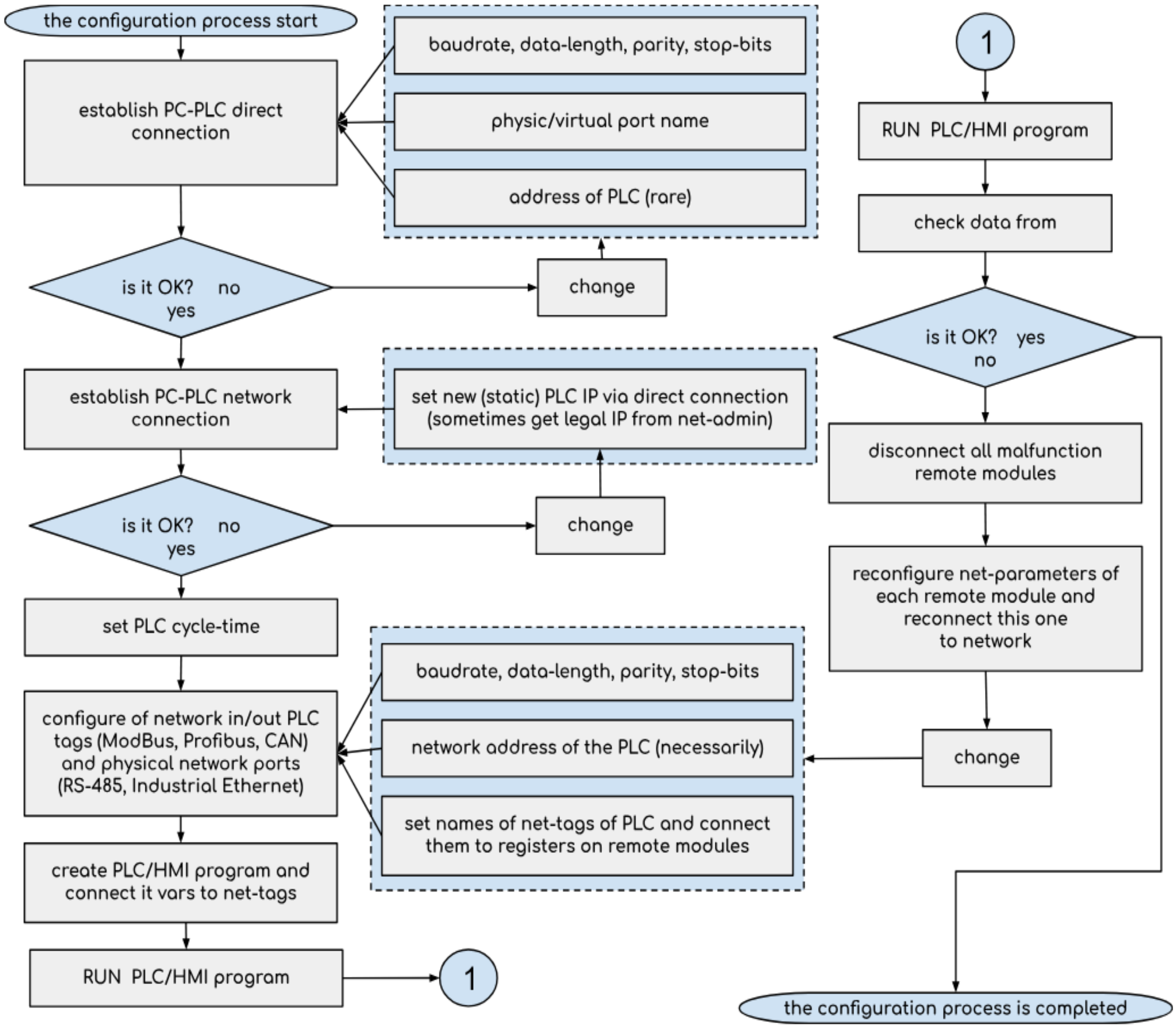

Figure 3: The algorithm of configuration and programming of the automated control network system

According to the list of selected equipment, the algorithm for software debugging of the system elements can be represented as follows (see figure 3):

1. Set the parameters of the PC-PLC connection via Ethernet or via other alternative connection (requires the involvement of a software product according to programming environment, creating a project in it for the controller program).
2. Set the value of the minimum and maximum PLC cycle-time (also performed in the programming environment).
3. Configure the network input PLC tags for operation by ModBus, Profibus, CAN and physical ports RS-485, Industrial Ethernet for communication with external modules and sensors.
4. Configure the network output PLC tags for operation by ModBus, Profibus, CAN and physical ports RS-485, Industrial Ethernet for communication with external modules and sensors.

5. Configure the network input/output PLC tags to work by industrial field bus to communicate with HMI devices (tags will pass values to the HMI panel for display on the screen, other variables will read values from the status registers of control buttons and digital input fields on the HMI screen).
6. Use the tools of standard libraries to compile a program in one of the IEC61131-3 languages to control the parameters of the technological process.
7. With the help of configurator programs of external input modules to set the configuration parameters of physical outputs, determine their names and network addresses in industrial field buses.
8. With the help of configurator programs of external output modules to set the configuration parameters of physical outputs, determine their names and network addresses in industrial field buses.
9. Use the HMI panel configurator program to create a control process visualization form. Create links between display elements/values and program PLC tags. Setting up the HMI panel to work in the field bus with the appropriate field protocol.

The specificity of typical network systems of field-level automation is that the equipment is uplinked during the operation over Modbus, Profibus, CAN. The panel, PLC, I/O modules have RS-485 ports and, with a simple two-wire connection, create a digital network which can operate without external control. The connection takes place using network tags in each device and under the control of a master device, such as a PLC. However, if the enterprise network case is considered with remote control from SCADA, the PLC and the SCADA-installed computer are connected to Ethernet. A connection established between them involves setting configuration parameters of network devices. If the connection is established, the interaction with the process flows through special variables in the PLC and OPC server, which is software connected to SCADA by tags.

4.2. Experiment

Experiments in education can be provided in groups of students that are getting qualification in the field of automation. The standard ways to involve proposed methods in the study process are accepted. There were such approaches: applied the proposed system of study material to academic disciplines of the last year and year before the bachelor degree achievement was completed (2020-2021, 2021-2022 academic year). The brief description of some parts of study sequences that were learned in and out of the classroom and performed by students in the laboratory practices as well as a brief description of the laboratory hardware array are presented below.

Description of basic equipment

To perform educational experiments, a branch was developed on the basis of laboratories equipment of the Department of Automation and Computer-Integrated Technologies of Zhytomyr Polytechnic State University. CoDeSys-compatible industrial PLC was used. Analog/discrete input/output is realized by PLC-compatible individual modules. The controller is located distantly from the power equipment. The control signals will be transmitted via the Modbus industrial network from the controller to the analog output module, and the signals from the sensors will be transmitted in the same network via the analog input module. Only I/O modules allocated nearby PLC. A modern HMI with color display, touch control capability and advanced program for interface creation are used. The HMI programming system which provides typical operations is also connected to the field network via the Modbus protocol. USB/RS-485 and USB/RS-232 interface converters are used to ensure PC versatility with system modules. We will consider the automated control system, where the PLC will play the role of PID controller with input and output signals, which will be implemented through network I/O modules.

Setting the configuration according to proposed method

Several step to realize experiment are shown as follow:

1. Setting the PLC connection parameters over the Ethernet network (requires the involve- ment of the CoDeSys software product and the creation of a project for the controller program):

- Run CoDeSys IDE and start a new project / set PLC model / create Program Organization Unit (POU), type of POU set to Program; choose language of POU save the project to the local PC disk.

- The PLC must have its own IP in the local network. To download the program to the controller, communication is established with the PLC via Ethernet: Login Communication parameters New TCP/IP (level 2): select IP address and port (sometimes standard IP and controller port set by the manufacturer are offered). After that the connection to the controller is created (Login Login). It is now possible to download the program to the controller when it is ready. At this stage, it is still recommended to try to connect to the PLC to check the communication channel.

If the connection does not work, it is possible for the following reasons:

Reason 1: If access to the local network is not allowed to devices with an unregistered MAC address, you must contact the network administrator to provide a fixed IP to the controller with a specific MAC address.

Reason 2: After connecting to the controller, in case the controller IP is not fixed, it is possible to specify a new IP address for the controller. In our case, this can be done with the SetIP command, via the so-called PLC Browser. In other cases, it may be stored in hardware configuration tabs. It is important not to forget to enter the new IP in the parameters to the Communication parameters set, because after changing the IP, the connection to the old IP will be immediately terminated.

Reason 3: The IP should be chosen so that the computer and the controller are working on the same subnet. Therefore, if there is no connection, it is still possible to establish it via Ethernet by installing another IP of the computer itself. Configure computer network settings needs to open the properties of the current LAN - Ether- net connection and to set the additional IP address of the computer in the properties of the TCP/IP protocol in the settings of the Windows network environment.

2. Setting the values of the minimum and maximum PLC cycle times (performed in the CoDeSys IDE). Typical cycle limits for PLC can be MinCycleLength, ms = 15, MaxCycle- Length, ms = 100.

3. Specify the configuration of the network input tags of the PLC to work in the field bus to communicate with the network sensor input module. In our case it is ModBus.

The setting is to set the protocol status parameters when the PLC is running in the Modbus network (master/slave). By default, the Append Subelement parameter is selected as Modbus (Master) (see figure 4, a).

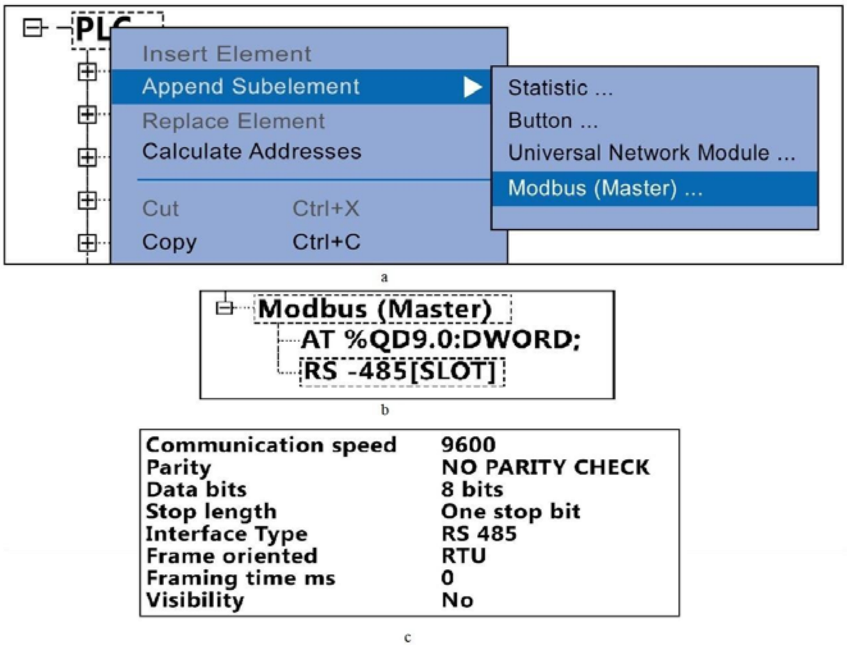

Figure 4: The structure of the menu of configuration of network interaction modules between PLCs and devices in the Modbus network: a) adding a network module directly; b) connecting the module to the PLC port; c) determining the parameters of network exchange.

That is, our PLC will act as the connection manager. For Modbus (Master) mode, add a software module with physical interface parameters (e.g. RS-485 [SLOT] (see figure 4, b)). Next it is needed to configure the physical interface settings. At this stage, the recommended values of network exchange parameters are determined (see figure 4, c). Here it is necessary to take into account the parameters of the network interfaces of those remote modules, I / O modules, to which we will try to connect. For example, for some network devices, the recommended connection speed is 9600 bits per second. Next you need to determine the type of Modbus protocol: RTU, ASCII. Other parameters can be left unchanged, leaving the recommended values for them. The default value of such a parameter as parity check is left as NO PARITY CHECK, for the parameter number of data bits (Data bits) we leave value 8 bits, stop bits (Stop length) - One stop bit, Framing time - 0 ms .

The feature of further configuration is to add a sub-element (Append Subelement ->Universal Modbus Device) (see figure 5, a) connection to each remote module and adjust the sub-elements parameters. In fact, we are adding a virtual remote Modbus device. The main parameter is the Modbus network address parameter of the module (remote device) to which we establish a connection (ModuleSlaveAddress). Other parameters are usually left unchanged (Work mode, Polling time, Visibility, Amount repeat, Byte sequence).

The parameters Module IP, TCP Port, Net Mode are important (see figure 5, b). These parameters are set for the application of the Modbus network via Ethernet, i.e. the Modbus protocol built into the TCP protocol. This concept is called Modbus TCP, then each module additionally receives its own IP addresses.

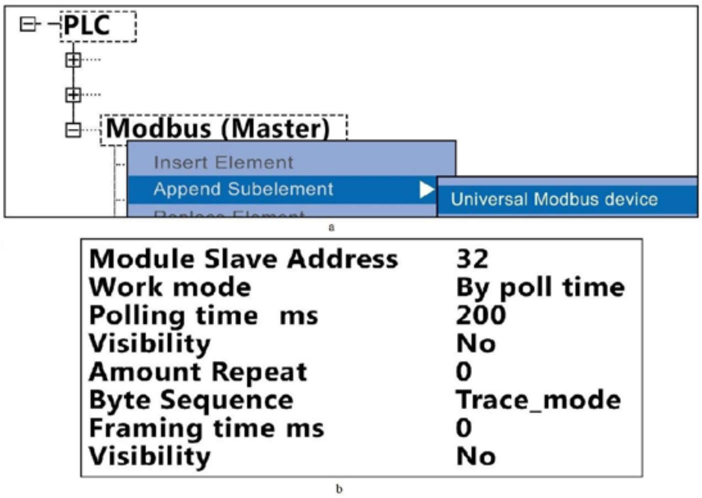

Figure 5: The structure of the menu for adding to the network interaction modules (between the PLC and the devices in the Modbus network) the unit of the network device to which the access will take place: a) the menu for adding the device unit; b) the window of network parameters of the added device.

Then the variables (registers) that need to be polled directly from this device are added to the network Modbus sub-element (actually a virtual remote Modbus device), and communication parameters are set for each of them. As the input module is configured, the corresponding configuration must be added as a separate sub-element (Real input module) (see figure 6, a). Next, specify the address of the register to read, which is assigned to register in the remote module (Register Address). At this address, our PLC will read the value to the input of the PID controller. The next important step is to define Modbus special actions, ie Modbus commands. Since the reading will take place, it is needed to use the commands "0x03" (read of one or more (up to 125 at a time) holding registers) or "0x04" (read of one or more (up to 125 at a time) internal registers) (see figure 6, b). It is also possible to use the command "0x02" (read of one or more (up to 2000 at a time) input coils), if there is a reading of discrete signals such as "on / off".

Finally, our register is now a full-fledged tag (variable) that we can use in a PLC program. The setting process must be completed by giving the register a name (for example, "pv_value", which under the form of a TAG will be connected to the input of the FBD block of the PID controller in the PLC program). Reading values from the remote module will take place without the direct involving of the PLC program, i.e. automatically.

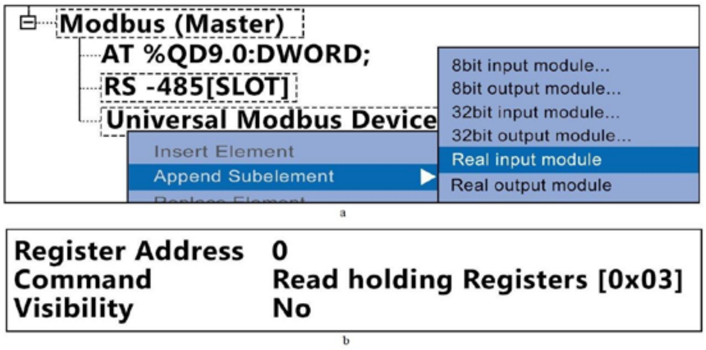

Figure 6: The structure of the menu for adding network interaction variables (between PLCs and devices in the Modbus network) to the network device block: a) menu for adding variables; b) the window of network parameters of the variable.

4. Setting the configuration of the PLC network output tags for operation in the field network, with aim that is communication with the network module for outputting control signals to power supply devices or flow control devices.

Here it can be used the already added network configuration element Modbus (Master) (see above). The only difference is that it is needed to write the data to a remote register now. So another Universal Modbus device is added. Similarly, it is set to Modbus address (ModuleSlaveAddress), and communication parameters (Work mode, Polling time, Visibility, Amount repeat, Byte sequence).

Next, it is necessary to add to the module the tag (register) in which our PLC will write data on the network. In fact, these are the physical outputs of the remote network signal output module. Therefore, guided by the already described above method, a separate sub-element is added - the register for writing values (Register output module). Similarly, the register is given the internal hardware address that it has in the remote network output module. The Modbus command is also selected to write the value to this register. Use the "0x10" command (Preset / Write Multiple Registers) or the "0x06" command (Preset / Write Single Register). It can also be the command "0x05" (Preset / Write Single Coil), or the command "0x0F" (Preset / Write Multiple Coils), if the control of digital output type "on / off". The configuration should be completed by providing a register name in order to use it as a tag (variable) in the PLC program (for example, change "pwm_mk", which will output the FBD block of the PID controller).

5. Configure the network input / output tags (variables) in the PLC program to work in the field network as a master to communicate with the HMI device. The tags created in this step will transfer the values from the PLC program to the HMI panel for display on the screen, other tags will read the values from the status registers of the control buttons and digital input fields on the panel screen and transfer them to the PLC program.

- The simplest configuration of the HMI panel starts with assigning register addresses that will store numeric values involved in the visualization form (they are usually set arbitrarily within the panel memory areas: volatile bit register memory - PSB, volatile double byte memory PSW , non-volatile PFW memory, or retentive registers, etc.). Bit registers are usually assigned to save the status of buttons, indicators, marks confirming the actions of the operator - the standard function of acknowledge. Byte registers are used to store numerical values of settings, measured values of technological parameters. Byte register sets - to store text notifications and data sets for graphic trends and logs.
- Next, in the programming environment of the operator panel, a new project of the presentation visual form is created, which will serve as the operator's work panel.
- Given that the panel will work in the field network in slave mode, it's not need to set special functions for the selected HMI registers. All it must be is set the slave mode, the network address of the HMI device in the HMI project settings, and write down the assignments and addresses of the selected registers operating inside the panel itself.
- Based on the data on the internal addresses of the HMI registers, it is need to return to the PLC programming environment (CoDeSys in our case), and create a network module in the main program (in our case, the network configuration element Modbus (Master)), which will communicate with

HMI, and for which it is need to specify the network address of the HMI. The next step is to add the appropriate network sub-elements (Register output module, Register input module) to the network configuration. Each subunit that is created must be configured to the internal HMI address of the register and the name of the corresponding tag for the PLC program. These sub-elements will be for reading or writing data using field network functions (for Modbus these are the functions: "0x02", "0x03", "0x04", "0x05", "0x06", "0x05", "0x06", "0x10", "0x0F"). For each of them specific HMI registers exist. This methodological stage is described more fully in the previous. For example, if a visual form of elementary PID controller configuration via the Modbus network had been created , the following tags and their processing are enabled: controller output power / tag name "pwm" / HMI address = PSW300 / PLC Modbus write function = 0x06; the current value of the technological parameter (stabilized by the regulator) / "pv_value" / "PSW301" / 0x03; the value of the PID-regulator setting point / "cur_sp" / "PSW302" / 0x03; bit of auto-adjustment status of the PID-regulator / "anr_state" / "PSB300" / 0x01.

6. Using the tools of standard libraries for a PLC-controller program creation.

Typically, PLC manufacturers provide well-designed software elements that perform standard control functions: a variety of relay-regulators, PID-regulators, signal processing units, analytical units. All this is delivered to the user in the form of software libraries for the PLC programming environment. In our case, the PID_Regulators library is among the list of CoDeSys libraries available for the project. Depending on the language of the IEC61131 standard, the PLC program will be created, the library software elements will be presented in proper form. This can be a representation in the form of FBD blocks, functions of the ST language, functions of the IL language, and so on.

Also in our case when the creation of the program is being it is considered that results of measurement of technological parameter should arrive in the software block of the PID-regulator from the external measuring device - the network input module, and control signals from the PID-regulator will be transferred to the external output module. These values can be received and transmitted using the network tags already defined previously.

The example of an FBD program for a PLC that performs PID control is shown in figure 7.

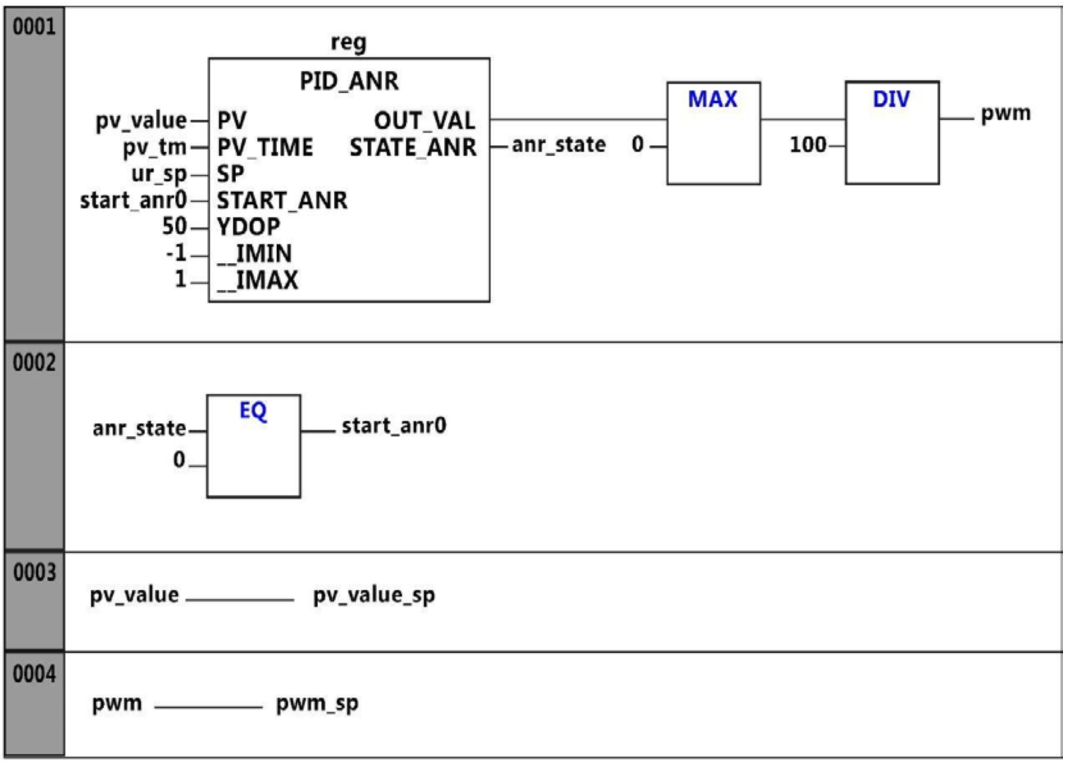

Figure 7: The example of the structure of PLC program that performs the functions of a remote network PID-regulator

Analysis of the residual level of knowledge

Analyses of the knowledge level in the area of field level were performed and could be shown by table 1 and diagrams on figure 8 and 9.

Table 1

Distribution of knowledge about the field level of automation systems between students of different years and transfer of knowledge between years of study

Academic year -->	**2020-2021**	**2021-2022**	
Categories -->	last year of bachelor's degree	year before bachelor's degree	last year of bachelor's degree
The qualification works with the mentioned “industrial network based automation system”	40%	—	70%
The individual projects works with the mentioned “industrial network based automation system”	60%	40%	50%
Tests successful results at the defined field of knowledge	90%	70%	90%
Tests quality level results at the defined field of knowledge	high: 40% middle: 50%	high: 50% middle: 20%	high: 60% middle: 30%
Region needs for professionals in a certain field of knowledge on the labor market	greater than 30 persons	greater than 50 persons	greater than 50 persons
Successful professional implementation of bachelors in the labor market	30%	—	40%

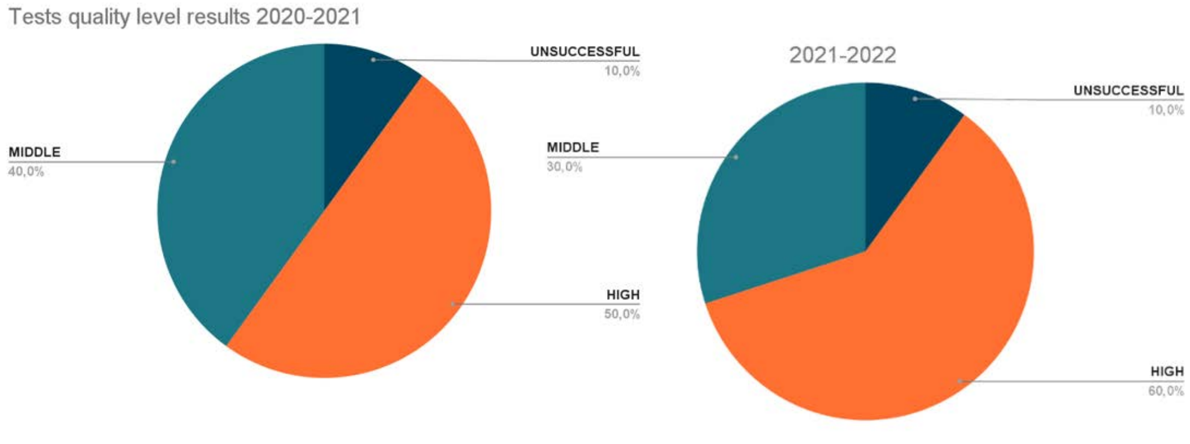

Figure 8: Of the graduate year students’ tests quality level results (a transfer of knowledge between years of study and a decent assessment of residual knowledge for students that are studying automation)

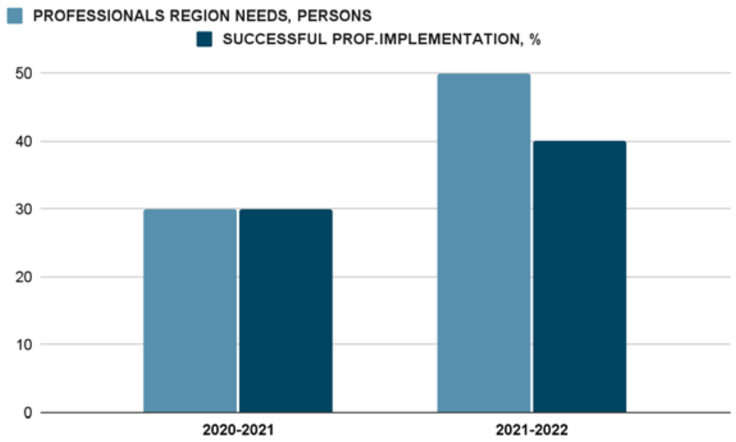

Figure 9: Successful professional implementation of bachelors in the labor market

5. Discussion

Achieved experimental results prove that implementation of a complex approach in early stages of the automatic systems study (that include field level of automation systems) can be realized into improvement of the professional knowledge perception skills on the later stages of study. This process can be defined as transfer of knowledge between years of study. Fair to point out that the students, who got special knowledge in the early stages, became understand the employers' requirements much early and, thus, have a good motivation in studying. The materials presented in the main part undoubtedly expand the view on the organization of training of future students of engineering specialties related to the development and implementation of automation systems at distributed facilities. The methodological basis embedded in this cluster of educational material is a "skeleton" that will allow building up structured parts clarifying certain aspects of concrete practical issues regarding the development of distributed network systems of automatic control.

The material, which until now was represented by scattered pieces of technical recommendations, instructions, user guides, and references for specialist engineers, has now been presented as a single unit, parts of which are structured, a selective sequence of teaching has been made, and new connecting parts have been added to this material. Now this set of educational content can be considered the basis for further expansion and addition of the educational methodology of distributed automation systems engineering. Although the presented methodology is open for revision and is not the last resort.

In the experimental section, an example of educational design is given, where we can clearly trace the relationship between all technical (hardware and software) elements of the system. The inseparability of the flow of information, which the student must learn to organize and configure, is shown. It is important that no component can be removed from the educational process, otherwise it will not be possible to teach how to build a complete system of automation, the feature of which is its networking and distribution of functions and components.

6. Conclusions

The materials presented in the article can be used to apply research of the empirical level of the studying process of automatics engineers and professionals everyday practice. The results of research consist of a systematic learning process of the configuration and software design for automated control systems on field-level technological processes with a typical hardware structure. The areas of knowledge that need to be used to perform the main stages of the synthesis of the control system are described. This list of requirements is proposed as a separate input stage of the synthesis. The recommended general structure of elements and connections of the modern control system is offered, which according to the list of components and performed functions can be applied to the small and medium complexity automated systems. The stage of interface configuration and programming is represented by the developed list of benefit stages of synthesis of the software part of the control system.

Specific and detailed implementation of recommendations have been developed for each stage. The confirmed experimental results of technique application for basic elements of a system are archived in educational laboratories. The experimental results are also general and fully relate to the work on the creation of any automated control systems.

Approbation of the methodology of the sequence of presentation of the material and its content showed a good effect of knowledge transfer between years of study and a decent assessment of residual knowledge for students studying automation. The proposed methods, techniques and approaches can be used to improve teaching methods and STEM tools for students who are mastering in automation and computer-integrated technologies at the bachelor's and master's levels. In addition, a survey of employers shows a confident increase in demand for field-level automation system specialists.

Finally, note, the methodology leads to the specification of skills and abilities acquired by students, future specialists in the field of automation and machinery building, focusing on working and improving the performance and functionality of existing automated process control systems based on network technologies and PLCs, including in the context of Industry 4.0.

It is likely that the role of specialists in field-level automation systems will continue to grow, which will pose new challenges to the technical and engineering education system in today's global world.

7. References

[1] A. Sattar, Q. Bakhsh, M. Sharif, Industrial automation and manufacturing systems: Concepts and applications, in: Manufacturing Engineering, volume 903 of Advanced Materials Research, Trans Tech Publications Ltd, 2014, pp. 291–296. doi: 10.4028/www.scientific. net/AMR.903.291.

[2] N. Kumar, S. C. Lee, Human-machine interface in smart factory: A systematic literature review, Technological Forecasting and Social Change 174 (2022) 121284. URL: https: //www.sciencedirect.com/science/article/pii/S0040162521007186. doi: 10.1016/j.techfore.2021.121284.

[3] J. Mellado, F. Núñez, Design of an iot-plc: A containerized programmable logical controller for the industry 4.0, Journal of Industrial Information Integration 25 (2022) 100250. URL: https://www.sciencedirect.com/science/article/pii/S2452414X21000492. doi: 10.1016/j.jii.2021.100250.

[4] A. Schumacher, W. Sihn, Development of a monitoring system for implementation of industrial digitalization and automation using 143 key performance indicators, Procedia CIRP 93 (2020) 1310–1315. URL: https://www.sciencedirect.com/science/article/pii/ S2212827120305606. doi: 10.1016/j.procir.2020.03.012, 53rd CIRP Conference on Manufacturing Systems 2020.

[5] P. Negirla, R. Drut, ă, I. Silea, Availability improvements through data slicing in plc smart grid networks, Sensors 20 (2020). URL: https://www.mdpi.com/1424-8220/20/24/7256. doi:10.3390/s20247256.

[6] Y. Bobalo, M. Seniv, V. Yakovyna, I. Symets, Method of reliability block diagram visualization and automated construction of technical system operability condition, in: N. Shakhovska, M. O. Medykovskyy (Eds.), Advances in Intelligent Systems and Comput- ing III, Springer International Publishing, Cham, 2019, pp. 599–610.

[7] V. Alcácer, V. Cruz-Machado, Scanning the industry 4.0: A literature review on technologies for manufacturing systems, Engineering Science and Technology, an International Journal 22 (2019) 899–919. URL: https://www.sciencedirect.com/science/article/pii/ S2215098618317750. doi: 10.1016/j.jestch.2019.01.006.

[8] Y.-B. Zhao, X.-M. Sun, J. Zhang, P. Shi, Networked Control Systems: The Communication Basics and Control Methodologies, Mathematical Problems in Engineering 2015 (2015) 1–9. URL: https://ideas.repec.org/a/hin/jnlmpe/639793.html. doi:10.1155/2015/639793.

[9] J. Hugé, C. Mac-Lean, L. Vargas, Maturation of sustainability in engineering faculties – from emerging issue to strategy?, Journal of Cleaner Production 172 (2018) 4277–4285. URL: https://www.sciencedirect.com/science/article/pii/S0959652617315895. doi: 10.1016/j.jclepro.2017.07.143.

[10] K. F. Mulder, Strategic competences for concrete action towards sustainability: An oxy- moron? engineering education for a sustainable future, Renewable and Sustainable Energy Reviews 68

(2017) 1106–1111. URL: https://www.sciencedirect.com/science/article/pii/S1364032116002732. doi:https://doi.org/10.1016/j.rser.2016.03.038.

[11] P. L. Hardré, C. Ling, R. L. Shehab, M. A. Nanny, H. Refai, M. U. Nollert, C. Ramseyer, E. D. Wollega, S.-M. Huang, J. Herron, Teachers learning to prepare future engineers: A systemic analysis through five componments of development and transfer, Teacher Education Quarterly 45 (2018) 61–88. URL: https://www.jstor.org/stable/90020315.

[12] R. A. Fenner, D. Morgan, The Barcelona declaration revisited: core themes and new challenges, in: EESD2021: Proceedings of the 10th Engineering Education for Sustainable Development Conference, Building Flourishing Communities, University College Cork, Ireland, 2021. URL: http://hdl.handle.net/10468/11592.

[13] Holovnia, V., Horodyskyi, M., Tkachuk, A., 2021. Students training for numerical control machines programming by means of computer-aided manufacturing tools. *Journal of Physics: Conference Series*, 1840(1), 012043. DOI: 10.1088/1742-6596/1840/1/012043

[14] T. Hesketh, Control education in Australia for electrical engineers: Current status, future trends, IFAC Proceedings Volumes 33 (2000) 19–23. URL: https://www.sciencedirect.com/science/article/pii/S1474667017378321. doi:https://doi.org/10.1016/S1474-6670(17)37832-1, 5th IFAC/IEEE Symposium on Advances in Con- trol Education (ACE 2000), Gold Coast, Queensland, Australia, 17-19 December 2000.

[15] I. Volegzhanina, V. Adol'f, S. Chusovlyanova, Methodology for the design of didactic tools for the future transport engineers' competencies formation on the ontology basis, Transportation Research Procedia 63 (2022) 1743–1751. URL: https://www.sciencedirect.com/science/article/pii/S2352146522004446. doi:https://doi.org/10.1016/j.trpro. 2022.06.189, x International Scientific Siberian Transport Forum — TransSiberia 2022.

[16] M. Doyle-Kent, E. Watson, D&i in engineering education, past, present, and future: A country case study of ireland., IFAC-PapersOnLine 54 (2021) 564–569. URL: https://www.sciencedirect.com/science/article/pii/S2405896321019467. doi: 10.1016/j.ifacol.2021.10.509, 20th IFAC Conference on Technology, Culture, and International Stability TECIS 2021.

[17] Y. Umeda, J. Ota, F. Kojima, M. Saito, H. Matsuzawa, T. Sukekawa, A. Takeuchi, K. Makida, S. Shirafuji, Development of an education program for digital manufacturing system engineers based on 'digital triplet' concept, Procedia Manufacturing 31 (2019) 363–369. URL: https://www.sciencedirect.com/science/article/pii/S2351978919304214. doi: 10.1016/j.promfg.2019.03.057, research. Experience. Education. 9th Conference on Learning Factories 2019 (CLF 2019), Braunschweig, Germany

[18] Lorido-Botran, T. and Bhatti, M.K., 2022. ImpalaE: Towards an optimal policy for efficient resource management at the edge. *Journal of Edge Computing*, 1(1), pp.43–54. Available from: https://doi.org/10.55056/jec.572

[19] Lobanchykova, N.M., Pilkevych, I.A. and Korchenko, O., 2022. Analysis and protection of IoT systems: Edge computing and decentralized decision-making. *Journal of Edge Computing*, 1(1), pp.55–67. Available from: https://doi.org/10.55056/jec.573

[20] Vakaliuk, T.A., Spirin, O.M., Lobanchykova, N.M., Martseva, L.A., Novitska, I.V. and Kontsedailo, V.V., 2020. Features of distance learning of cloud technologies for the organization educational process in quarantine. *Journal of Physics: Conference Series*, Volume 1840, 012051, XII International Conference on Mathematics, Science and Technology Education (ICon-MaSTEd 2020) 15-17 October 2020, Kryvyi Rih, Ukraine. DOI: 10.1088/1742-6596/1840/1/012051