doi:10.22306/am.v6i4.81

Received: 12 Nov. 2021
Revised: 02 Dec. 2021
Accepted: 21 Dec. 2021

SOFTWARE APPLICATION FOR A SYSTEM WITH A PROGRAMMABLE LOGIC CONTROLLER

Jaroslav Romančík
Technical University of Kosice, Faculty of Mechanical Engineering, Letna 9, Kosice, Slovak Republic, EU,
jaroslav.romancik.2@tuke.sk (corresponding author)
Michal Kelemen
Technical University of Kosice, Faculty of Mechanical Engineering, Letna 9, Kosice, Slovak Republic, EU,
michal.kelemen@tuke.sk

Keywords: PLC, HMI, control system, ladder

Abstract: The article deals with the solution of an application for a building security system, which is solved using a programmable logic controller. A ladder program design for this application is created, and an application for the HMI touch screen is also designed. The application is tested on a simulation model in a software development environment. A training station system was used for practical prototype tests, which includes a programmable logic controller, HMI touch screen, profinet switch, power supply and other accessories for testing purposes.

1 Introduction

The gradual development of human society has also been reflected in the expansive development of modern technologies, which have become an integral part of our lives. We got used to most of the gains so quickly that we took them for granted. This fact does not only affect ordinary computers, mobile phones or other white goods. It works similarly in industrial engineering.

Without the use of automation elements, production, as we know it today, would not be possible. It would not be possible to produce so many products in such a short time, with such precision, if all these tasks were to be done by man. We have to admit that machines are more efficient. But still, there has to be a person at the top. In this particular case, the PLC programmer. As PLC (acronym from Programmable Logic Controller) is a frequently used tool for controlling robotic arms, conveyor belts and entire lines, it has also become a major player in mechatronics.

The implementation of PLCs began in the 80s of the 20th century, but their origin dates back to the 70s. The first company to address this issue was General Motors. In 1968, it announced a competition, the task of which was to build a control system as a replacement for the hitherto used relay, the reprogramming ability of which depended on the design and electronic connection. If changes were to be made in the activities, the cabling had to be changed, which was still often very confusing. Also, the switching of the relay is not infinite and has a limited number of cycles. Therefore, finding the relay that was not working caused a lot of maintenance problems.

Today, human activities are often replaced by machines. Since the trend is to mechanize and automate, a great means has proven to be PLC automation. We can meet him in every part of the industry, from the control of conveyor belts and robotic arms in the automotive industry, through the automation of filling and transfer machines in the food industry, to applications in smart homes from water heating to the opening of windows depending on the time of day. They are mainly used in places of industry where it is necessary to work with a large amount of information from sensors, as they offer a large number of inputs and outputs, whether digital or analog (Figure 1). Also, the PLC version is very advantageous, as they can be easily enclosed in a large box, called a rack, and so they can be placed in larger halls. In this way, they are protected from unauthorized access. Also, the entire main cabling can be stored in this way, which protects the entire system from moisture, dust, other electronic devices or vibrations.

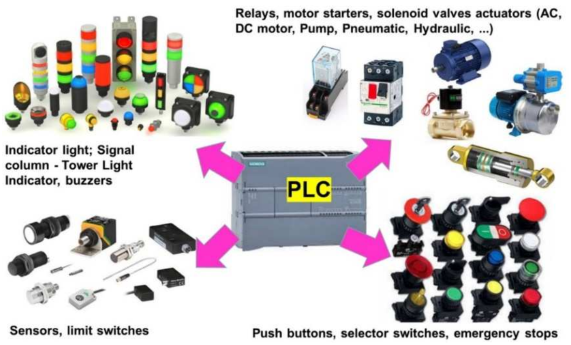

Relays, motor starters, solenoid valves actuators (AC, DC motor, Pump, Pneumatic, Hydraulic, ...)
Indicator light; Signal column - Tower Light Indicator, buzzers
PLC
Sensors, limit switches
Push buttons, selector switches, emergency stops

Figure 1 PLC in conjunction with sensors, actuators, controls and output modules

HMI touch screens (Human-machine interface - HMI panel), which are currently an integral part of the PLC, are also used together with the PLC. HMI touch screens are devices with sensitive touch screens for operator input and are usually mounted so that they can be operated by personnel or operator. They provide a visual graphical interface with up-to-date information about the controlled device and controls and allow easy operation and selection of device operation options [1-3].

2 Building security system

A security system is proposed for the complex, which will be monitored by staff (Figure 2). The complex has three possible inputs. During the day, people and employees enter the complex, but after a certain hour, when the staff activates the security system, no one will be allowed to enter the building without triggering an alarm. If someone still enters the building, an audible alarm will sound, accompanied by warning lights. The staff will also need to know the exact location of the breach in order to intervene properly. The created program will have to be executable and stoppable in the HMI display environment, as it should only work at a certain time of day. We also have to count on three entrances to the complex, the opening or closing of which we will simulate with the buttons. If the intruder enters the complex, this security breach will trigger the siren, which will simulate an audible alarm, and light up the LEDs on the training module, which will simulate the warning lights. The entire condition of the building will be monitorable using the HMI panel.

Figure 2 Building security system

In order to be able to work with PLC and HMI panel and to be able to create programs, it is necessary to set the device configuration in the environment where we create a program with these two devices and possibly add later modules if necessary to expand the device. The individual modules can be connected via the Profinet interface (Figure 3).

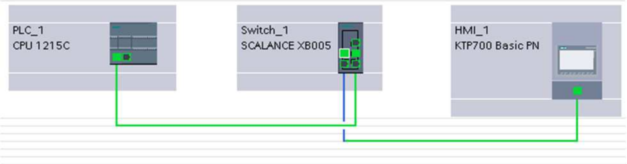


Figure 3 Topology of PLC and HMI devices

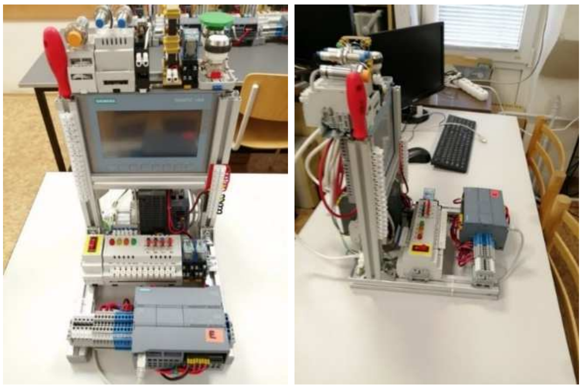


Figure 4 Training station with PLC and HMI panel

For prototype tests, it is then possible to use a training station with a PLC (Figure 4), which contains a PLC core module, HMI touch screen, industrial switch, power supply and other testing accessories such as switches, LED indicators, sensors, motors, solenoids, etc. [4-6].

3 Program design

The program is often implemented using the Ladder logic language, which is considered to be the simplest form of PLC programming. Ladder logic has evolved into a programming language that represents a program using a graphical scheme based on relay logic hardware circuit diagrams. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications. The name is based on the observation that programs in this language resemble ladders with two vertical rails and a series of horizontal bars between them. Whereas Ladder logic was once the only available entry for recording programmable controller programs, today, other forms are standardized in the IEC 61131-3 standard.

The solved alarm system should be switchable and switchable via the HMI panel. This function will be implemented using a switch. In order to make it clear what state the security system is in, this state will be indicated by the LEDs of the training module, as well as by the information on the HMI display. We will start with a Normally open contact, which we will call "start", which will be the input. We will store an assignment for it, which we will call "m_start" and it will be a memory variable that will be used for the HMI panel. We end this part with the output "alarm_functional", which in our case lights up the green LED diode. This part of the program will be used to turn on the security system. We will use a similar procedure to turn it off, but in this case we will use the Normally closed contact, but we will use the same input. By turning it off, we will switch the assignment "alarm_nofunctional", which will cause the red LED to light up (Figure 5).

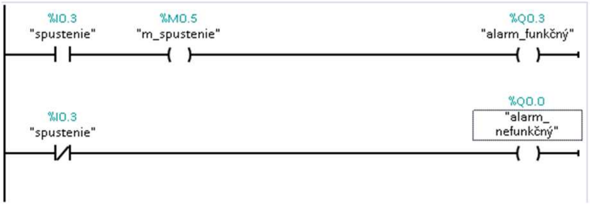


Figure 5 Ladder program - start and stop

In the next step, we will create the main part of the program. Because there are three entrances to the complex. We will monitor their opening and closing in our program using Normally open contact. We will create a separate branch in the program for each entrance. For easier orientation, we will call them, for example, "main_entrance", "secondary_entry", "garage_doors". Since we will also want to see their status on the HMI display, we will place an assignment as a memory variable for each input. In order to distinguish the individual memory variables and their assignment to the individual inputs, we will call them "m_main_input", "m_side_input" and "m_garage_doors". In order for our program to work properly and fully, it is necessary to realize that after the first alarm, the alarm must remain on, even if the entrance closes. We solve this problem using the logical operator Counter, specifically CTU (Count up), ie count up. After placing this block in Ladder, we can notice that this operator has its own inputs and outputs. CU serves as input, Q as output. PV determines the value by which the counter should count, the CV from which value it starts. R indicates reset. We substitute a value of 0 for CV, and a value of 1 for PV, which we achieve when the signal is applied by pressing the button to count the counter to 1, and even after releasing the button to keep the logic 1 at the output. If we want to reset it, we connect the "start" input to the R terminal. This will ensure that after the alarm is triggered, we reset it by turning off the security system, otherwise the alarm will remain triggered. We will use counters for each entrance. After Q we will add another memory variable, which will be needed when programming the HMI panel. We will end this whole part of the program with the "alarm" output, for which we will use the assignment operator, and it will be common for all three entrances. We will switch the siren on this operator. After that, we place the memory variable "m_alarm", which will display the alarm on the HMI display (Figure 6).

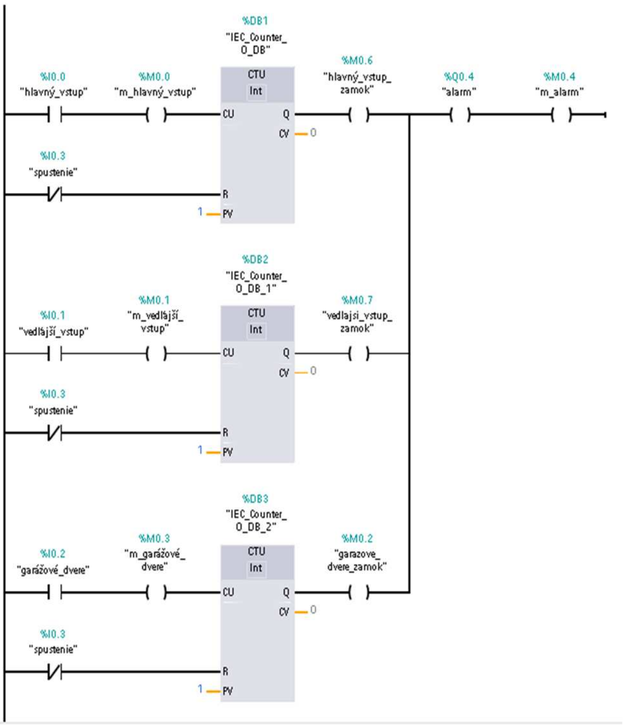


Figure 6 Ladder program - the main part of the program

The last part of the Ladder program will be the creation of signalling by means of LED diodes, announcing that the alarm has been triggered. For better signalling of the alarm, we will design the program so that the two remaining LEDs (orange and yellow) flash. You will need to use timers to blink. We start the program with the "alarm" output, which ended the previous part of the program, but this time we will use the contact normally open. From this step, it is clear that the flashing will be triggered, and only if the alarm is triggered. We will create two branches for individual LEDs. We start each branch with a TON timer. If we look at the timer, we find the designation IN, which means input. Q indicates output. PT and ET set the start and end time by how long the input signal is to be delayed. In this case, we will use a time of 2 to 0 seconds. In the first branch we add a timer identical to the previous one, but not in the second branch. This step will cause one LED to be shifted by 2 seconds, compared to the other. If we did not add this timer to our program, both LEDs would flash at the same time. Next we will add outputs for individual LED diodes. We will call them "warning_light_1" and "warning_light_2". They will be followed by other timers, with identical parameters as the previous ones. We end the program with RT (reset timer) operators, two for each branch. These operators have the task of resetting the timers, so that the LEDs in the loop flash. We will only reset the timers before and after the diodes, so we will not reset the first timer in the branch with three timers, as its task was only to shift the flashing frequency of one of the LEDs once (Figure 7).

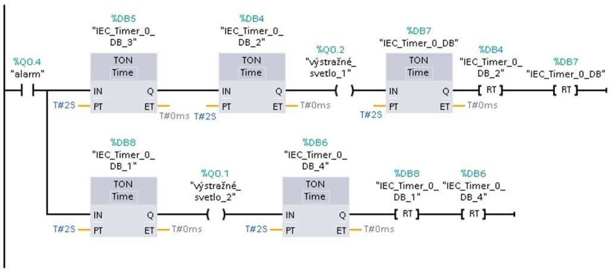


Figure 7 Ladder program - flashing LEDs

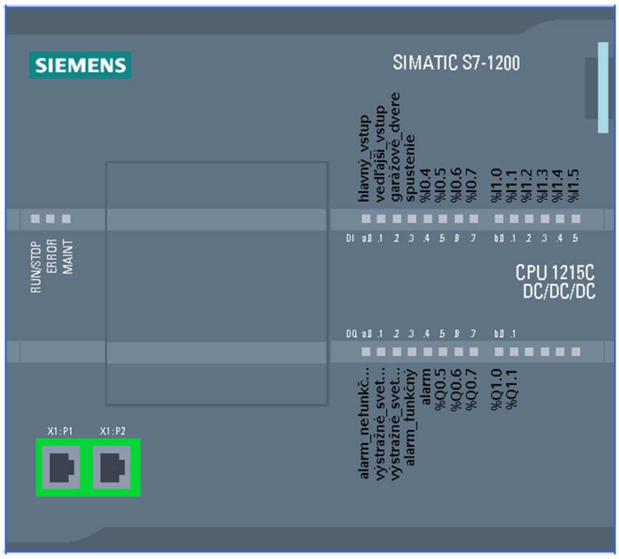
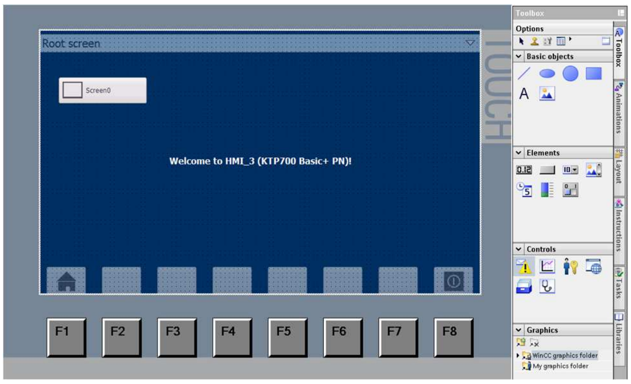


Figure 8 Virtual digital twin after program creation

After creating the entire Ladder program, we can notice that all the inputs and outputs we use are displayed on the virtual digital twin of our PLC (Figure 8). This display can help us to connect inputs and outputs using wires between the PLC and the training module. The next step will be loading the program into the PLC.

4 Program design for HMI panel

In the next step we will create a program for the HMI panel. For this particular solution, the appearance of the HMI with a dark blue background was chosen, with a home and sub screen and two buttons, one to turn off the program and the other to return to the home screen of the HMI panel (Figure 9).



Welcome to HMI_3 (KTP700 Basic+ PN)!

Figure 9 HMI design

We will create a graphical design of the HMI interface by placing graphical objects and their subsequent connection with events and visualization conditions via HMI tags connected to PLC tags. The visibility of individual objects can be solved using the functions Dynamize colors and flashing and Make visibility dynamic (Figure 10).

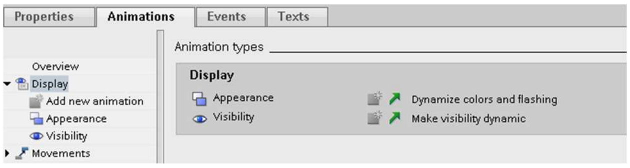


Figure 10 Visualization of objects

The first function of the program should be the ability to turn the security system on or off. Therefore, in our case, the inscription "INACTIVE" was created. It is made up of two parts. The word NO is visible in the logical state zero of the "HMI_start" tag, but one is invisible in the logical state. The same tag is also used for the second part of the word "ACTIVE". But in this case, it is red at logical zero, and green at logical unit. In this way, personnel can see the status of the security system not only by means of LED diodes, but also by a colored sign. Since he has to see where the violation occurred, we will create a table called "PLACE OF INFRINGEMENT". It contains three columns, namely "MAIN ENTRANCE", "SIDE ENTRANCE" and "GARAGE DOOR", with lock icons in front of each sign. They will be controlled by the tags: "HMI_main_entry_input", "HMI_current_entry_input", "HMI_garazove_door_lock". If the logical state is zero on them, they will be invisible. However, if the object is disturbed, the panel will display the label, depending on where the alarm was triggered. "ALARM" is invisible unless an alarm is triggered. If it has occurred, the inscription will appear and flash alternately red and yellow. The beacon above the word "ALARM" is made up of three identical images that overlap, are layered, and are white, red, and yellow. If the alarm is not triggered, a white beacon is visible, red and yellow are invisible. However, when the alarm is triggered, the white beacon disappears, becomes invisible, but the red and yellow beacons are visible. The yellow beacon, placed in the top layers, also flashes. This creates the illusion that when the alarm is triggered, the off white beacon will flash alternately red and yellow. In this way, the staff will be clearly informed that the complex has been disrupted. The beacon and the inscription "ALARM" are controlled by the logical state of the tag: "HMI_alarm". We will rename the "Screen" button, which will take us to the new screen, by a more appropriate name. In our case, the inscription: "GUARDED OBJECT" (Figure 11).

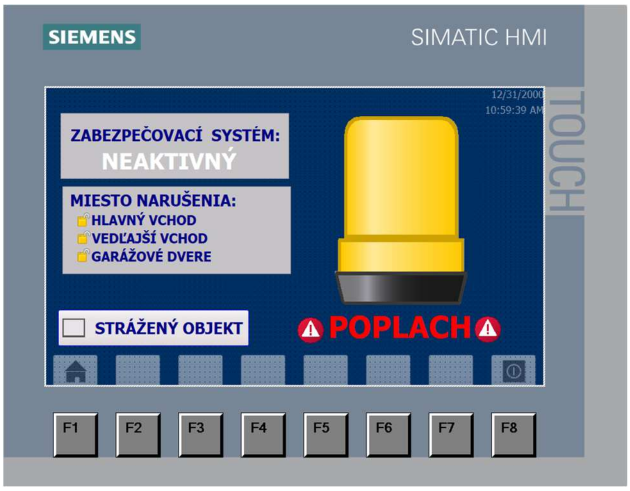


Figure 11 Graphic design HMI- Root screen

In the next step, we will create a graphical display of the screen, which we will be able to access using the "GUARDED OBJECT" button. On this screen, staff will be able to see the entire complex with the exact location of the individual entrances to the complex. He will also see the door open and close during the day, whether or not the security system is active.

Using images of buildings and the outside environment, we will create the appearance of a complex that will always be visible. We will add pictures of the door. We choose two for each entrance, when the door is open and when it is closed. We program the image of closed doors so that at logic state zero, the image is visible, but at logic state one is invisible. We program the image of the open entrance in reverse. After stacking them on top of each other and placing them in the appropriate place of the complex, the illusion of opening and closing the door arises. We will use this procedure for all entrances. They will be controlled by the tags: "HMI_main_input", "HMI_side_input" and "HMI_garage_doors".

As personnel see the current state of the door, they must know where the alarm was triggered, even though the intruder managed to close the door. Therefore, we will add a table with the names of the entrances along with the lock icons and a warning triangle with an exclamation mark above each door. If the security system is active and the complex is disturbed, a warning symbol will appear above the specific entrance and will flash, regardless of whether the door is closed or open. The lock icon will only be visible if the door has been opened. Their function will be controlled by the same tags that we used to program the table "DISTRIBUTION LOCATION", namely: "HMI_main_input_input", "HMI_additional_lock_input", "HMI_garazove_door_locks". At the bottom of the screen is the "HOME" button. This button takes you back to the home page (Figure 12).

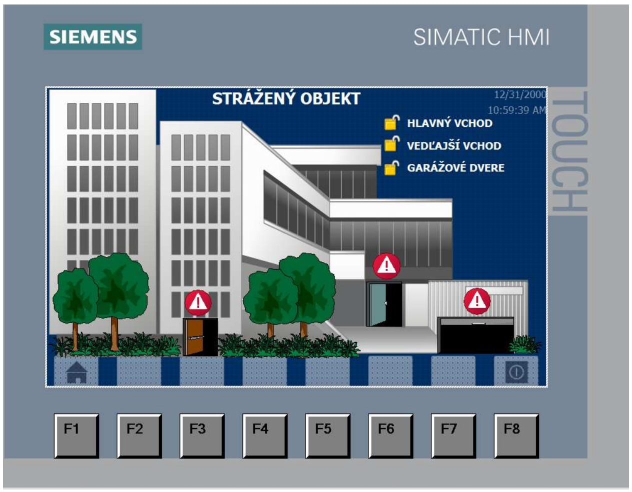


Figure 12 Graphic design HMI- Guarded object

As in the case of loading a program for a PLC, the program for the HMI panel must first be compiled, and then we will perform a loading into the HMI panel device. After loading we can test its functionality. After a successful simulation in the program environment, the implementation into the training station and the successful testing of the proposed program were also successful.

5 Conclusion

PLC systems are located in switchboards together with other electrical modules and are not normally accessible to ordinary users. For communication with the user, the PLCs are equipped with a Human-machine interface (HMI), which allows the user to control the status of the controlled system and can intervene in the control process if the situation in the controlled process so requires.

PLC control systems are installed in devices with high electrical voltage, where there is a risk of electric shock and thus only the technician who has the appropriate authorizations in connection with the safety of work with electrical equipment has direct access to the hardware of PLC control systems. In many cases, the controlled system is a technological system, the costs of which are high, so it is important that only authorized persons work with the PLC control system [7-17].

Acknowledgement
The work has been accomplished under the research project KEGA 030TUKE-4/2020 financed by the Slovak Ministry of Education.

References

[1] LANDER, A.: *Programmable Logic Controllers: The Evolution of a Disruptive Technology*, Engineering.com, [Online], Available: https://new.engineering.com/story/programmable-logic-controllers-the-evolution-of-a-disruptive-technology [26 May 2021], 2021.

[2] SIMATIC S7-1200 *Getting started with S7-1200*, [Online], Available: https://cache.industry.siemens.com/dl/files/875/39644875/att_76194/v1/s71200_getting_started_en-US_en-US.pdf?download=false [26 May 2021].

[3] SIMATIC S7-1200 *Programming Guideline for S7-1200/1500*, [Online], Available: https://assets.new.siemens.com/siemens/assets/api/uuid:c7de7888-d24c-4e74-ad41-759e47e4e444/Programovani-S7-1200-1500-2018.pdf [26 May 2021], 2021.

[4] KELEMEN, M., SINČÁK, P.J.: Programmable logic controller training stands for educational purposes, *Technical Sciences and Technologies*, Vol. 21, No. 3, pp. 274-280, 2020.

[5] KELEMEN, M., LIGUŠOVÁ, J., LIGUŠ, J., MAXIM, V.: Additional Modules for Programmable Logic Controller Based Training Stands, *Technical Sciences and Technologies*, Vol. 22, No. 4, pp. 170-176, 2020.

[6] KELEMEN, M., KELEMENOVÁ, T., VIRGALA, I., PRADA, E., MIKOVÁ, Ľ., HRONCOVÁ, D., VARGA, M., SINČÁK, P.J.: *Training stations with programmable logic controllers*, Technological Forum 2021: 12th International technical conference: 30.6. - 02.07.2021, Jaroměř, ČR, pp. 117-121, 2021.

[7] VITKO, A., JURIŠICA, L., KĽÚČIK, M., MURÁR, R., DUCHOŇ, F.: Embedding Intelligence into a Mobile Robot, *AT&P Journal Plus*, Vol. 2008, No. 1, Mobile robotic systems, pp. 42-44, 2008.

[8] BOŽEK, P.: Robot path optimization for spot welding applications in automotive industry, *Tehnicki vjesnik / Technical Gazette*, Vol. 20, No. 5, pp. 913-917, 2013.

[9] DUCHOŇ, F., BABINEC, A., KAJAN, M., BEŇO, P., FLOREK, M., FICO, T., JURIŠICA, L.: Path planning with modified a star algorithm for a mobile robot, *Procedia Engineering*, Vol. 96, pp. 59-69, 2014.

[10] PÁSZTÓ, P., HUBINSKÝ, P.: Mobile robot navigation based on circle recognition, *Journal of Electrical Engineering*, Vol. 64, No. 2, pp. 84-91, 2013.

[11] ABRAMOV, I.V., NIKITIN, Y.R., ABRAMOV, A.I., SOSNOVICH, E.V., BOŽEK, P.: Control and Diagnostic Model of Brushless DC Motor, *Journal of Electrical Engineering*, Vol. 65, No. 5, pp. 277-282, 2014.

[12] KONIAR, D., HARGAŠ, L., ŠTOFAN, S.: Segmentation of Motion Regions for Biomechanical Systems, *Procedia Engineering*, Vol. 48, pp. 304-311, 2012.

[13] VIRGALA, I., MIKOVÁ, Ľ., KELEMEN, M., HRONCOVÁ, D.: Snake-like robots, *Acta Mechatronica*, Vol. 3, No. 4, pp. 7-10, 2018.

[14] MIKOVÁ, Ľ., VIRGALA, I., KELEMEN, M.: Embedded systems, *Acta Mechatronica*, Vol. 3, No. 2, pp. 1-5, 2018.

[15] KELEMENOVÁ, T., FRANKOVSKÝ, P., VIRGALA, I., MIKOVÁ, Ľ., KELEMEN, M., DOMINIK, L.: Educational models for mechatronic courses, *Acta Mechatronica*, Vol. 1, No. 4, pp. 1-6, 2016.

[16] LIPTÁK, T., KELEMEN, M., GMITERKO, A., VIRGALA, I., HRONCOVÁ, D.: the control of holonomic system, *Acta Mechatronica*, Vol. 1, No. 2, pp. 15-20, 2016.

[17] KURYŁO, P., PIVARČIOVÁ, E., CYGANIUK, J., FRANKOVSKÝ, P.: Machine vision system measuring the trajectory of upper limb motion applying the Matlab software, *Measurement Science Review*, Journal of Institute of Measurement Science of Slovak Academy of Sciences, Vol. 19, No. 1, pp. 1-8, 2019.

Review process

Single-blind peer review process