Lighting Engineering & Power Engineering
2022, Vol. 61, No. 3, 107–121
https://doi.org/10.33042/2079-424X.2022.61.2.03

OPEN ACCESS

Electric Drive SCADA Development with Vijeo Designer (Schneider Electric)

Oleksandr Aksonov[1], Vladyslav Pliuhin[1], Yevgen Tsegelnyk[1], Oleksii Slovikovskyi[2], Oleksii Duniev[3], and Andrii Yehorov[3]

[1]O. M. Beketov National University of Urban Economy in Kharkiv, Kharkiv, Ukraine
[2] National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine
[3] National Technical University Kharkiv Polytechnic Institute, Kharkiv, Ukraine

Article History
Received:
23 December 2022
Accepted:
28 December 2022
Published online:
30 December 2022

Keywords
Human Interface Device; SCADA; Electric Drive; Vijeo Designer; Control Panel; Control System; Remote Control

Abstract
The paper is devoted to the disclosure the construction of the SCADA system for managing the operating modes and monitoring the electric drive of the overhead crane. The material is provided in the form of step-by-step instructions for building such a system in the Vijeo Designer program. The creation of a new project, interface elements, binding variables with interface elements, setting the behavior of dynamic interface elements are shown in detail. The issue of communication of global variable visual elements of the interface with the M241 controller and the program in SoMachine has been revealed. The developed interface allows to control three mechanisms of the bridge crane: main hoist, auxiliary hoist and carriage movement. It is shown how to control frequency converters that are connected to the controller through different types of communication: direct analog to the controller, and through the Ethernet/IP protocol. The work will be useful for academic purposes in teaching students as well as for researchers and industrial engineers.

INTRODUCTION

Scientific and technological progress is impossible without industrial automation. With the development of microprocessor technology and the growth of computing power of computers, it becomes possible to simulate various processes without the participation of the object itself.

As proof of this, we can cite the concept of combining intelligent SCADA software with the object-oriented agent structure Industry 4.0 [3, 4], considered in [1, 2]. Thus, they seek to replace existing manual systems with fully self-controlled processes in order to improve the entire production system. This optimizes production time by reducing user intervention in the overall process [5].

To implement this new trend, it is necessary to establish control over every process related to production in all industries. The Supervisory Control and Data Acquisition (SCADA) system allows you to establish communication through the OPC-UA server, which provides data exchange with control devices such as programmable logic controller (PLC) and programmable automation controller (PAC), etc. [6, 7]. The SCADA system allows the user to access monitoring, control and automation equipment through several protocols [8] and is a three-layer architecture. It allows real-time simulation of an industrial process starting from the control level, then the process control level and finally the field instrument level [9]. Each level has its own set of tools, controllers and modules.

For the control (scheduling) level, the LabVIEW development environment is used with an appropriate set of tools for data logging and supervisory control [10–12].

For the process control level, it is possible to use industrial programmable logic controllers (PLC) from Siemens (Simatic series) [13, 14] or Schneider Electric (Modicon, Twido series) [15, 16] and the corresponding software package.

The field device level requires a process simulation module.

Corresponding author: vladyslav.pliuhin@kname.edu.ua (Vladyslav Pliuhin)

Cite as: Aksonov, O., Pliuhin, V., Tsegelnyk, Y., Slovikovskyi, O., Duniev, O., & Yehorov, A. (2022). Electric drive SCADA development with Vijeo Designer (Schneider Electric). *Lighting Engineering & Power Engineering, 61*(3), 107–121. https://doi.org/10.33042/2079-424X.2022.61.3.04

The use of information modeling [17] and digital tools [18] is becoming increasingly important in the activities of leading industrial enterprises. At the same time, the leading role is given to improving communication between modern equipment and maintenance personnel using digital means. Virtual commissioning can significantly reduce commissioning time, error rates and costs [19]. However, the industry continues to have difficulty integrating these new technologies [20].

In the industrial automation environment, special attention is paid to the human-machine interface (HMI) system [21, 22]. This is a user interface that connects a person to a machine or device, or a screen that allows the user to interact with the device [23].

In [24], the impact of artificial intelligence on the growth of achievements and technological breakthroughs during the transition from Industry 4.0 to Industry 5.0 is analyzed.

Most human-machine interfaces are configured using software on computers. Applications are developed using the Vijeo Designer software [25], which is included in the SoMachine package. In addition to the basic interface settings, it allows to use the built-in multimedia capabilities and the web interface [26].

The dialog interface includes all functions necessary to control the operation of a machine or plant. Depending on the requirements, the operator can stop and start the operation of the installation, monitor and make adjustments necessary for the regular operation of the system.

This article shows the development of a modeling system that works based on a combination of two programs: SCADA and Vijeo Designer. The first software calculates the parameters of the various operating conditions of the drive, and the second is used to simulate the human-machine interface.

As a result, a training device can be developed that allows users to gain hands-on experience with the drive or be used as a tool to evaluate and improve the skills of existing substation operators.

CREATING AND CONFIGURING A PROJECT IN VIJEO DESIGNER

The touch panel project is developed in a separate Vijeo Designer program made by Schneider Electric and involves solving the following tasks:

- development of a graphical interface for interaction with the user (operator);
- connection of graphic elements of the panel with variables of the SoMachine project;
- establishing communication between the touch panel project and the controller.

The appearance of the panel interface, which is the final goal of the paper, is shown in Fig. 1–4.

Vijeo Designer can be opened as a stand-alone application. However, for the synchronous development of the panel and the controller and subsequent simultaneous simulation of the two devices, the Vijeo Designer program must be opened directly from SoMachine.

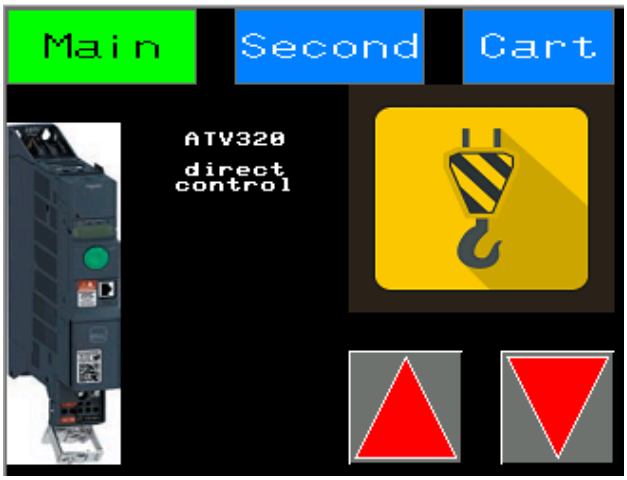

Figure 1. Touch panel window: main hoist

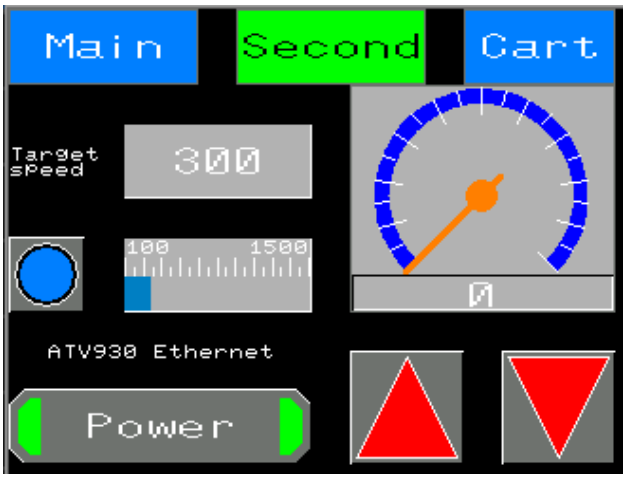

Figure 2. Touch panel window: secondary hoist

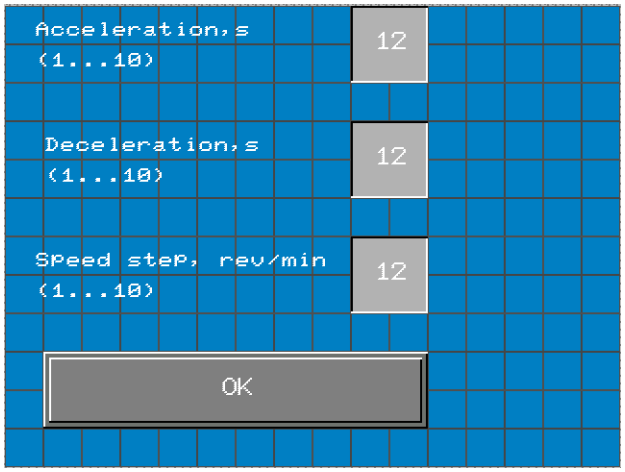

Figure 3. Window for setting drive parameters

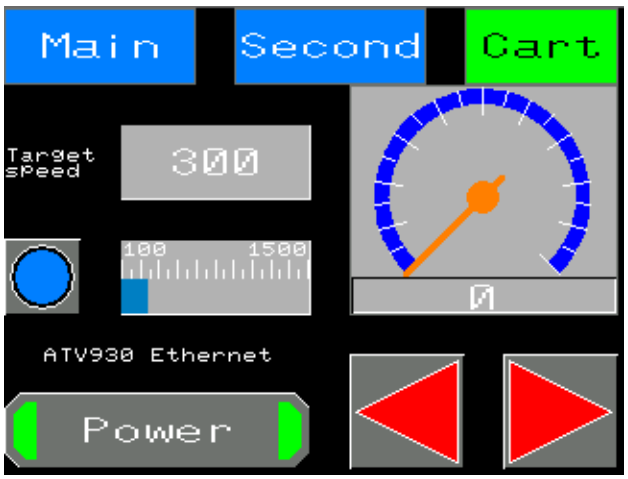

Figure 4. Touch panel window: cart movement

To establish communication between the controller and the panel, return to SoMachine and double-click the MyController (TM241CE24T/U) line. In the window that opens, in the empty Nodename field, enter the service name of the controller, which was automatically assigned and is located in the table in the Nodename column (tab of the controller page "Controller selection") (Fig. 5). Or one can also simply double-click on the Nodename in the table.

Copy this name (DESKTOP-N7TQ98P in the example) to the clipboard and return to Vijeo Designer.

Expand the line "IO Manager", then "SoMachineNetwork01" and click on the line "SOM_HMISTU855". In the "Network Equipment Configuration" window that appears, paste the name of the controller node from the clipboard into the Equipment Address or Node Name field and click the "OK" button (Fig. 6). Now, finally, let's move on to the development of the graphic interface of the HMISTU855 touch panel. Let's create blanks for five panels. This should be done for the first, as the panels will contain buttons with links to each other. In the Navigator object navigator, right-click on the "Base Panels" line and select the "New Panel" item in the context menu. After the performed operations, the navigator window will have the appearance shown in Fig. 7.

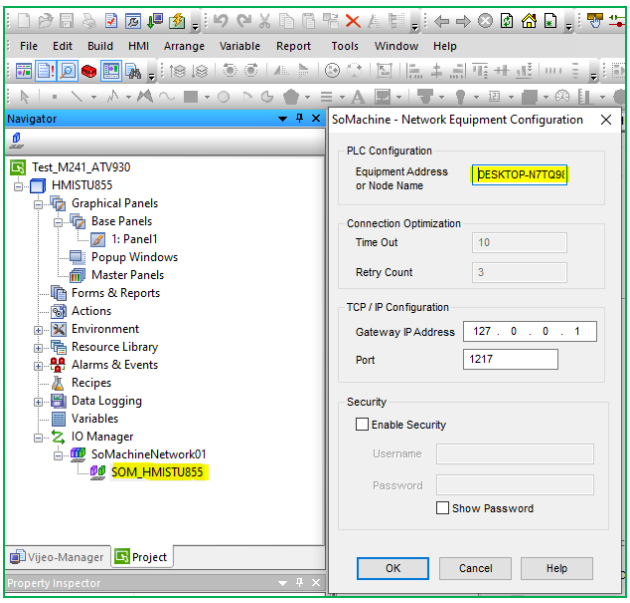

Figure 6. Configuring the controller address in Vijeo Designer

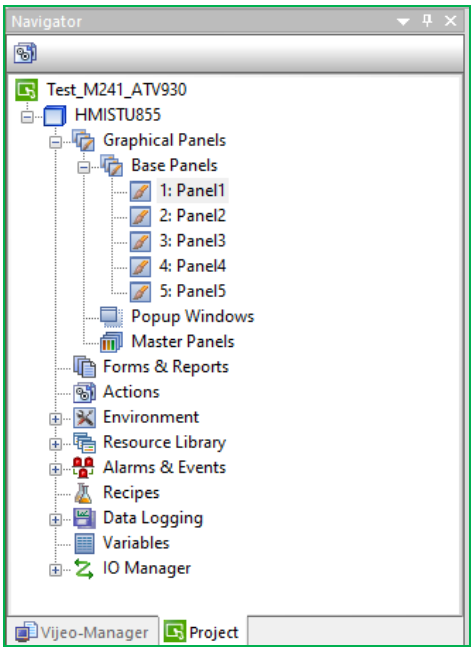

Figure 7. Navigator window with blanks for 5 panels

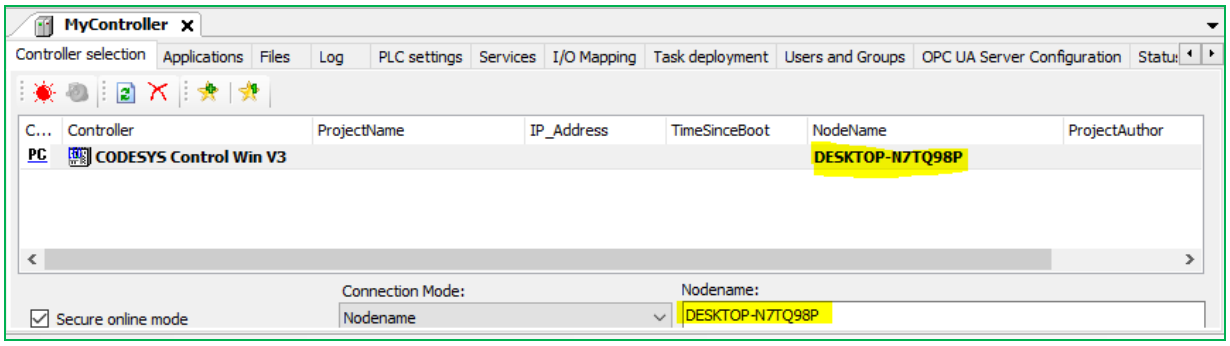

Figure 5. Designation of controller node name

Change the names of the panels as shown in Fig. 8 (right click on the name of each panel, select the "Rename Panel" item in the context menu and enter a new name). Explanation of panel assignments:

- PanelMain – main lift drive control
- PanelSecond – control of the auxiliary lift drive;
- PanelCart – cart drive control;
- PanelOptions1 – ATV930 frequency converter parameters;
- PanelOptions2 – parameters of the ATV930 frequency converter.

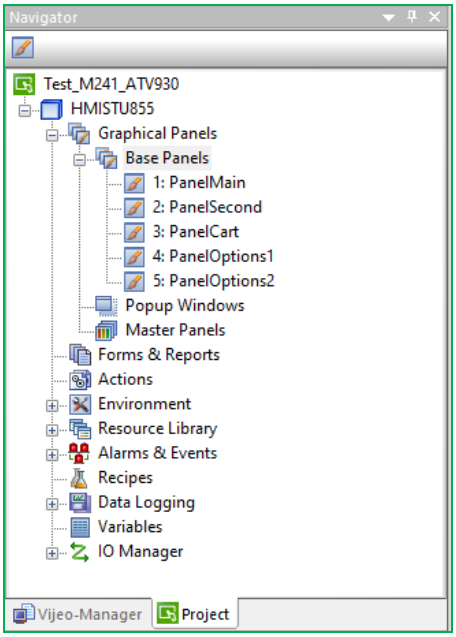

Figure 8. Navigator window with changed panel names

The touch panel exchanges data (values of global variables) with the controller. To provide access to these variables, they must be transferred from the SoMachine project to the Vijeo Designer project. To do this, go to the SoMachine program and open the Tools tree tab. Right-click on the Application line and select "Add Object", "Symbol configuration…" from the context menu.

In the window that appears, click on the "Add" button. After that, compile the project by clicking the "Build" button on the toolbar or directly on the Symbol Configuration page (Fig. 9).

In the list of available variables, highlight GVL and POU_Main in the empty rectangles. Export the selected variables: right-click on the Symbol Configuration line and select "Export Symbols to Vijeo-Designer" from the context menu.

The selected variables will now be available in the touchpad project in Vijeo Designer.

MAIN HOIST DRIVE PAGE INTERFACE

In the navigator, select the panel with the name PanelMain – an empty form of the panel will open. Click once on the Switch tool, then click an empty space on the shape. Without releasing LMB, stretch the dotted rectangle by two cells in height and five cells in width. In this case, the Switch element parameters window will immediately open (Fig. 10).

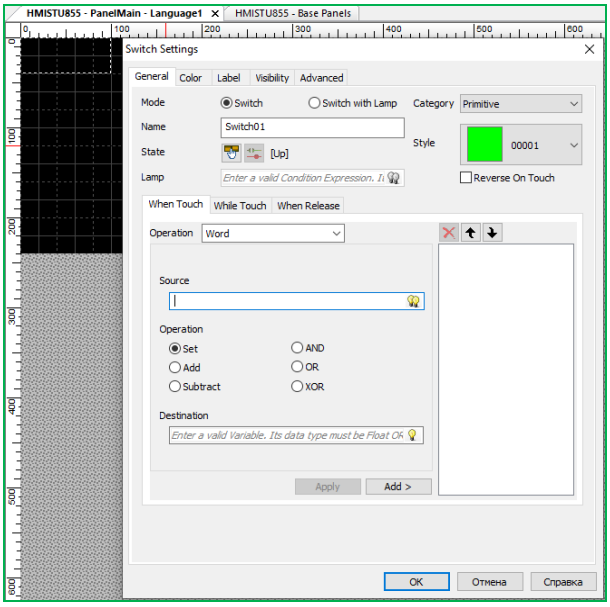

Figure 10. Switch element parameters window

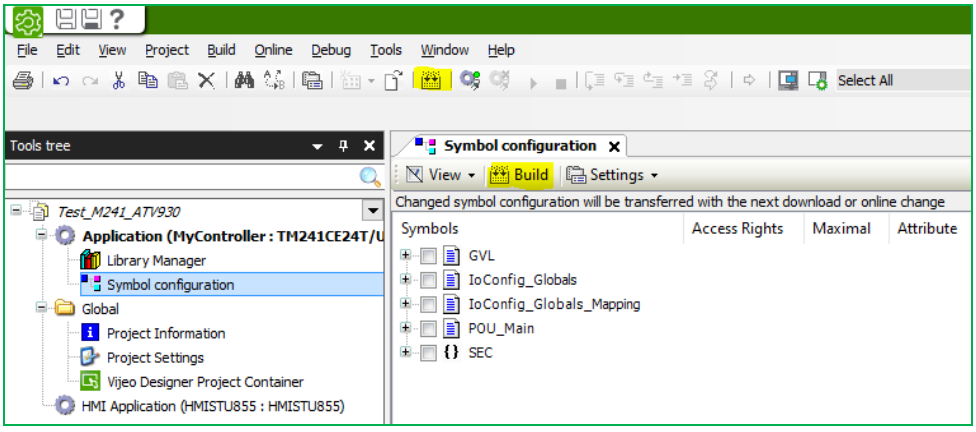

Figure 9. Compiling the project

Set Mode to Switch for the Lamp. Next to the Lamp parameter, click on the two yellow lamps - a variable selection window will open. In this window (Fig. 11), go to the lower SoMachine tab, expand the GVL list, select the xMain variable and click the "OK" button.

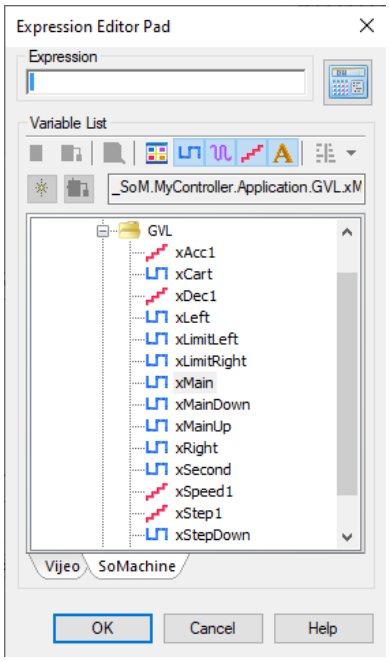

Figure 11. Variable assignment for a lamp

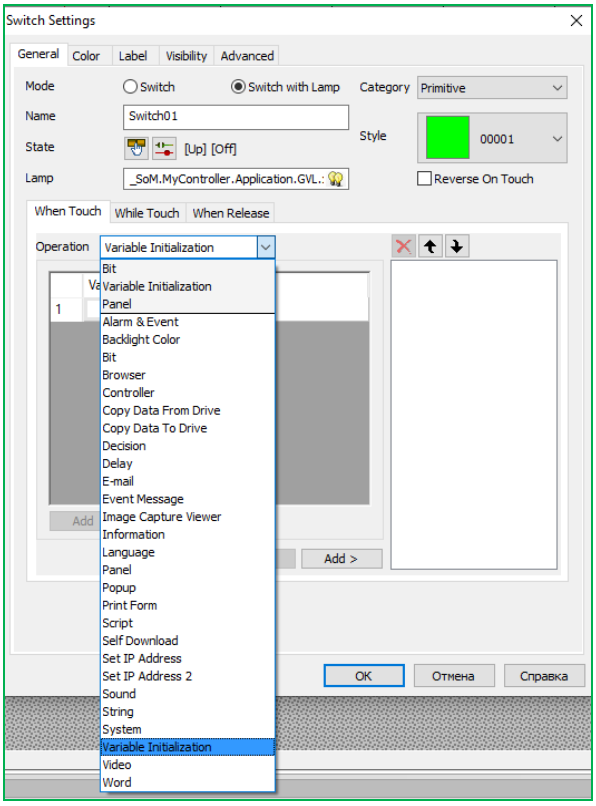

Figure 12. Adding a variable initialization function to a button

For the Operation parameter, open the drop-down list and select the "Variable Initialization" item (Fig. 12).We need the ATV320 frequency converter to activate on the PanelMain and the ATV930 output circuit contactors to open. In addition, the button with the label of the active panel should be green, and the others should be blue. For this, the xMain variable must be set to TRUE, and the xCart and xSecond variables must be set to FALSE. It is for this that initialization of variables is required when clicking on the button that is created on the PanelMain panel.

In the empty field of row number 1 click in the Variable column and then click on the yellow lamp that appears on the right side. In the window that opens, go to the SoMachine tab, expand the GVL list, select the xMain variable and click the "OK" button (Fig. 13). In the Value field next to the selected variable, enter the number 1.

Click on the "Add" button at the bottom of the table and repeat the described operations for the xSecond and xCart variables, only for them in the Value field set the value to 0 (Fig. 14).

To add completed operations to the stack of the Switch element, click on the "Add>" button at the bottom.

Open the Operation list again and select the "Panel" item. Click on the three buttons in the "Change Panel" section and in the window that appears, select the PanelMain line and click the "OK" button (Fig. 15). After that, click on the "Add>" button.

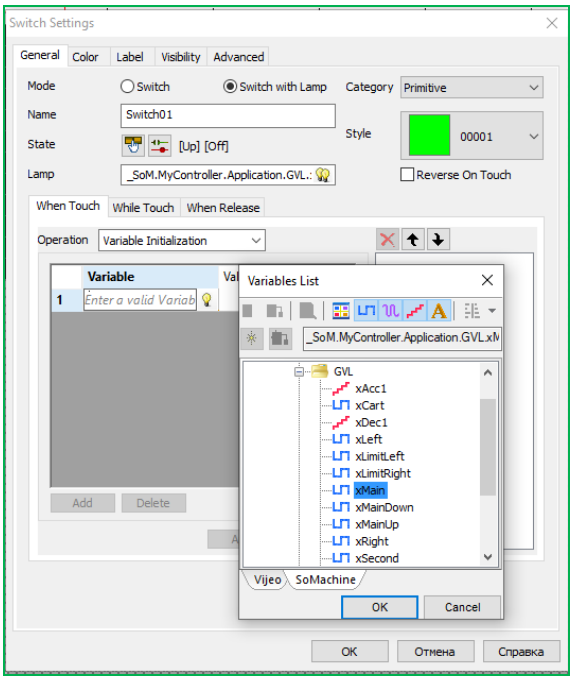

Figure 13. Selecting a variable for initialization

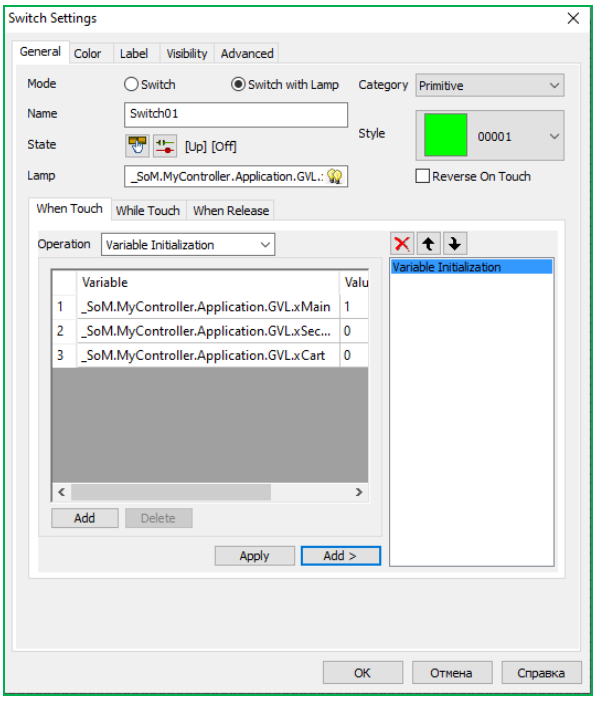

Figure 14. List of created variables for initialization

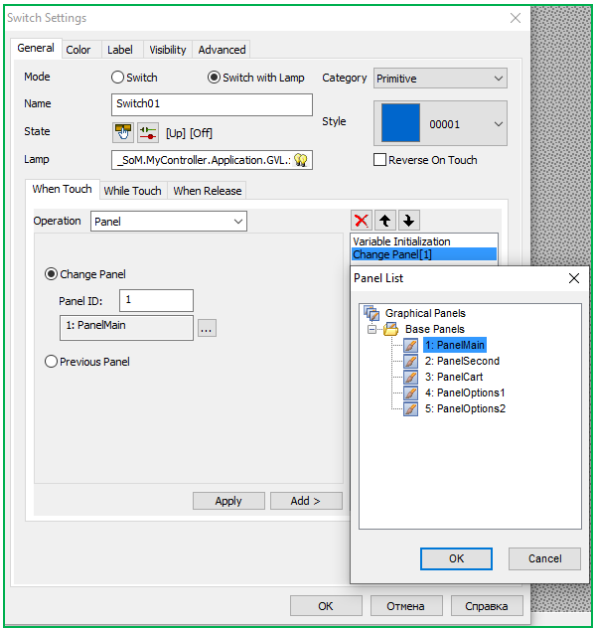

Figure 15. Binding a panel switch operation to a button

Go to the "Color" tab and select the colors for the active "On" and inactive "Off" button (Fig. 16).

Go to the "Label" tab. Set the Label Type to Static and enter the word "Main" in the text box. Choose the font (Font parameter) Vijeo Sans Serif with dimensions Font Width (width) and Font Height (height) 16 (Fig. 17).

This completes the work with setting the first button. Once again, make sure that all the above parameters are selected correctly and click on the "OK" button. The button shown in Fig. 18 will appear on the panel form.

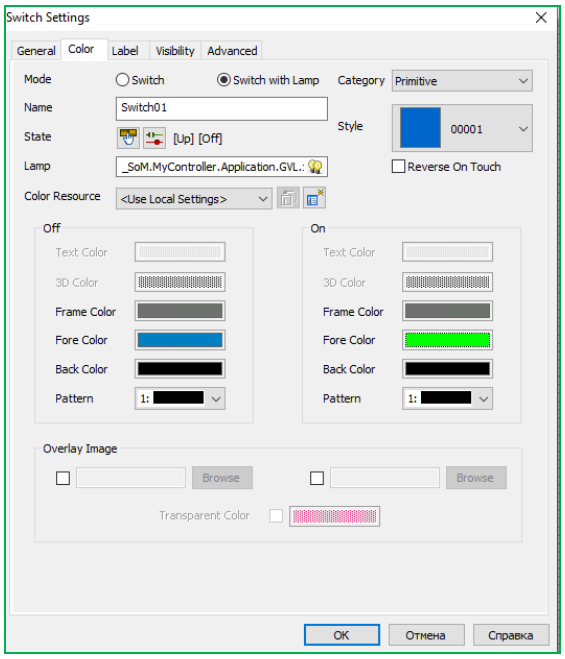

Figure 16. Setting the button color

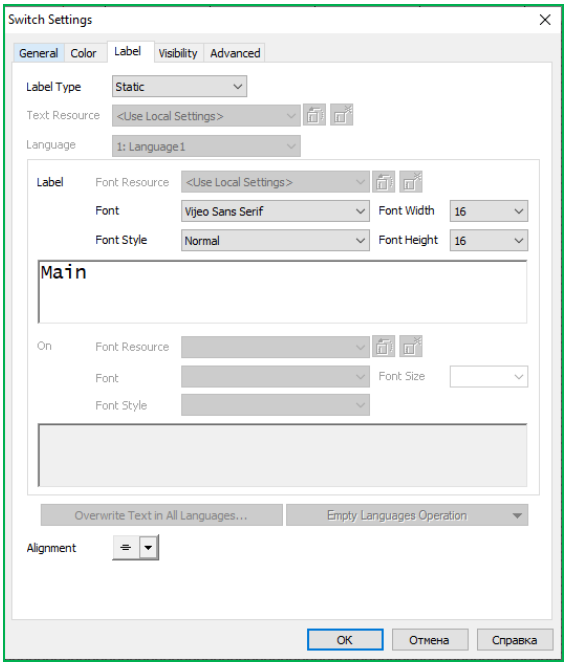

Figure 17. Assignment of the label for the button

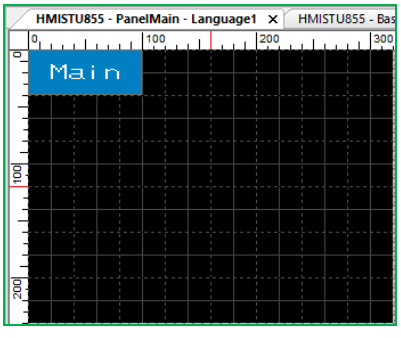

Figure 18. Form of the panel with the activation button of the ATV320 control page

Similarly, create buttons named Second and Cart. Second button options:

- the lamp is bound to the variable xSecond;
- in the initialization table xMain = 0, xSecond = 1, xCart = 0;
- the value of the Change Panel parameter: PanelSecond.

Cart button options:

- the lamp is bound to the xCart variable;
- in the initialization table, xMain = 0, xSecond = 0, xCart = 1.
- the value of the Change Panel parameter: PanelCart.

The form of the PanelMain panel will look as shown in Fig. 19. Now select all the buttons with a grid and copy them to the clipboard. Then, alternately, go to PanelSecond and PanelCart panels, click once on the form and insert buttons from the clipboard.

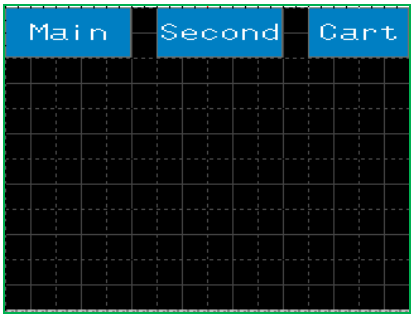

Figure 19. Form of the PanelMain panel with page switching buttons

Select the Image item in the menu and select a rectangle with a width of 3 cells and a height of 9 on the form. Select the image of the inverter in the browser and click "OK".

Select the Image item in the menu and select a rectangle with a width of 7 cells and a height of 6 on the form. Select an image of a crane hook in the browser and click "OK".

Select the Text item in the menu, select a rectangle on the form and enter the text “ATV320 direct control”. The result of the performed operations is shown in Fig. 20.

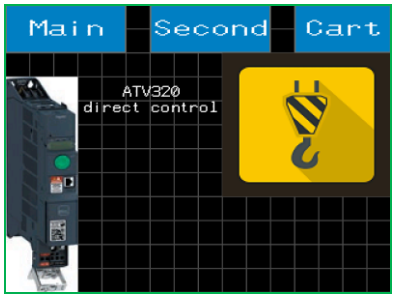

Figure 20. Form with images and label

Select the Switch item in the menu and select a rectangle with a height and width of 3 cells on the form. Select “Switch with Lamp” as the Mode parameter. For the Lamp parameter, set the xMainUp variable. In the list of Operation parameters, select Bit, the behavior of the bit is “Momentary ON”. For the “Destination” parameter, select the xMainUp variable and click the “Add” button. Expand the list of Style button styles, click “Browse” and select the “up” arrow button. The view of the General tab from button settings is shown in Fig. 21. The Color tab setting is shown in Fig. 22.

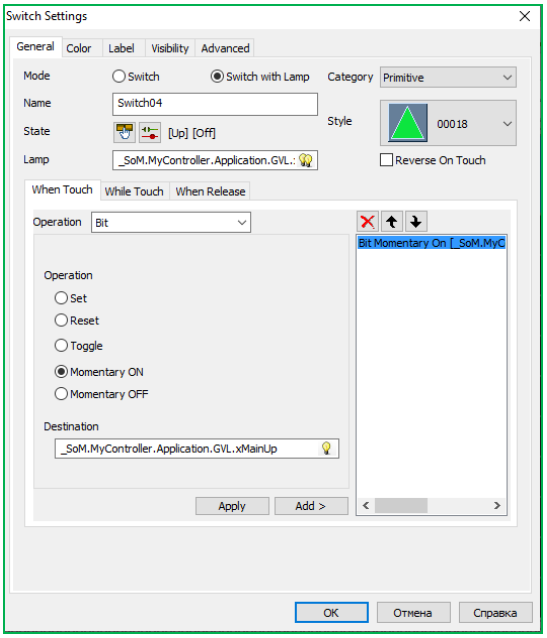

Figure 21. Configuring the “up” button behavior for the ATV320

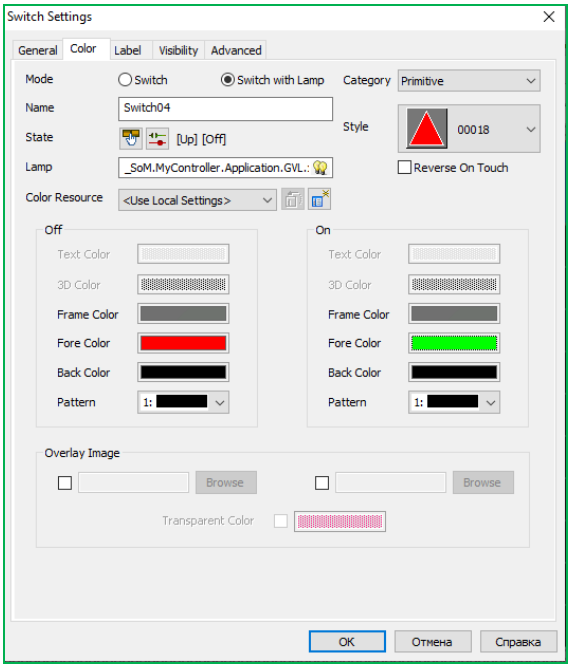

Figure 22. Up button color settings for ATV320

Similarly, create a button to move the load "down" by choosing the xMainDown variable for the Lamp and Destination parameters. The appearance of the fully completed PanelMain form is shown in Fig. 23.

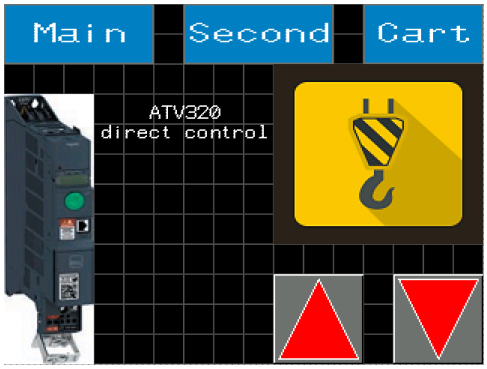

Figure 23. Drive Motion Control page from ATV320

SECONDARY HOIST DRIVE PAGE INTERFACE

Open the PanelSecond helper page form. Place the label "Target speed" on the form. Select the Numeric Display tool from the toolbar and select a rectangle 2 cells high and 5 cells wide on the shape. For the Variable parameter, select the variable xSpeed1, other settings are shown in Fig. 24.

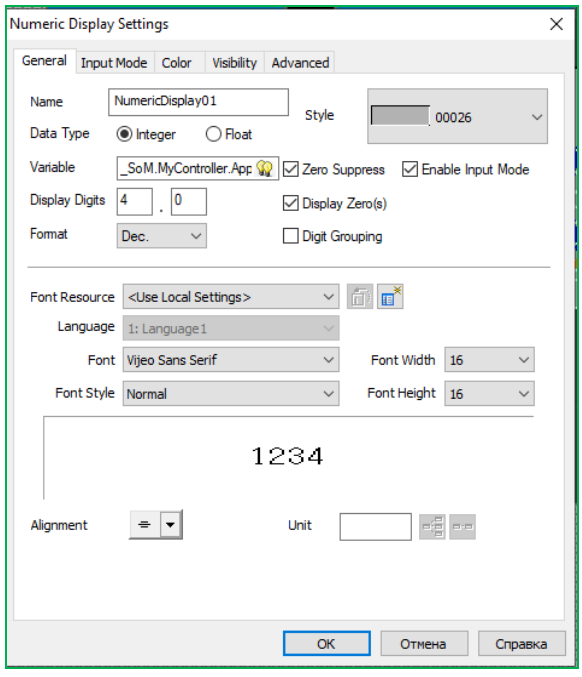

Figure 24. General tab of the Numeric Display element settings

On the Input Mode tab, check the "Enable Input Mode" option. Select the Meter tool from the toolbar and select a rectangle 6 cells high and 7 cells wide on the shape. For the Variable parameter, select the variable iActualVelo_ATV930, other settings are shown in Fig. 25.

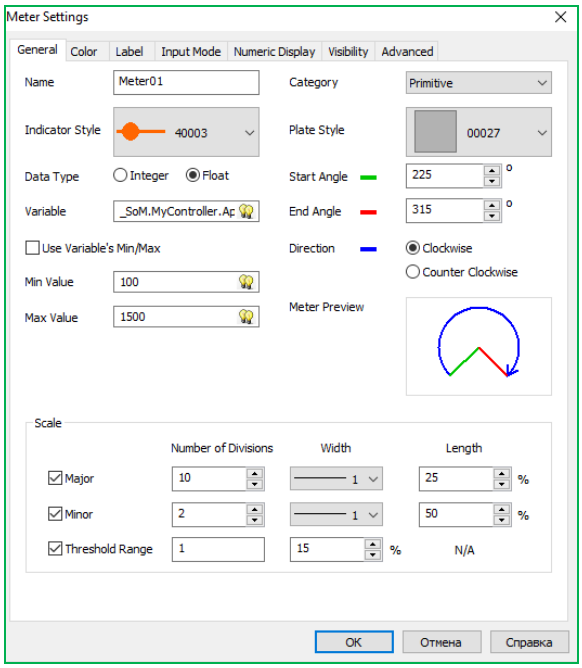

Figure 25. General tab of the Meter element settings

On the Numeric Display tab, check the Enable Numeric Display option. Enter the number 5 in the Display Digits field. The result of the performed operations is shown in Fig. 26.

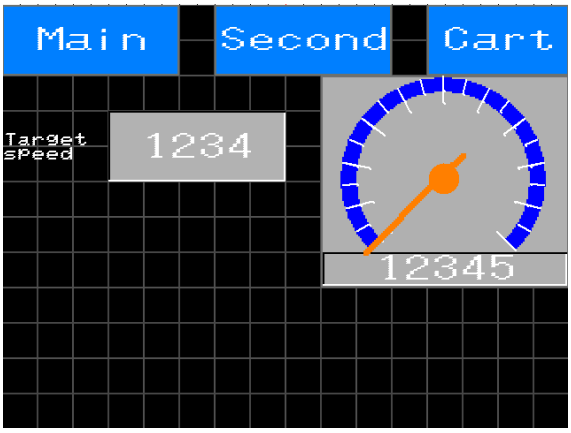

Figure 26. PanelSecond page form (intermediate stage)

Create a Horizontal Bar Graph element 2 cells high and 5 cells wide on the form. For the Variable parameter, specify the variable xSpeed1. On the Input Mode tab, select the "Enable Input Mode" check box. The controller settings are shown in Fig. 27 and 28.

Create an ATV930 transducer activation button with a height of 2 cells and a width of 8. For the Operation parameter, select Bit with the behavior Momentary ON, for the Lamp and Destination parameters, select the xCmdEn_ATV930 variable. On the Label tab, set the Label Type to Static. In the text box, type the word "Power" in Vijeo Sans Serif font size 16x16. The settings of the General and Color tabs are shown in Fig. 29.

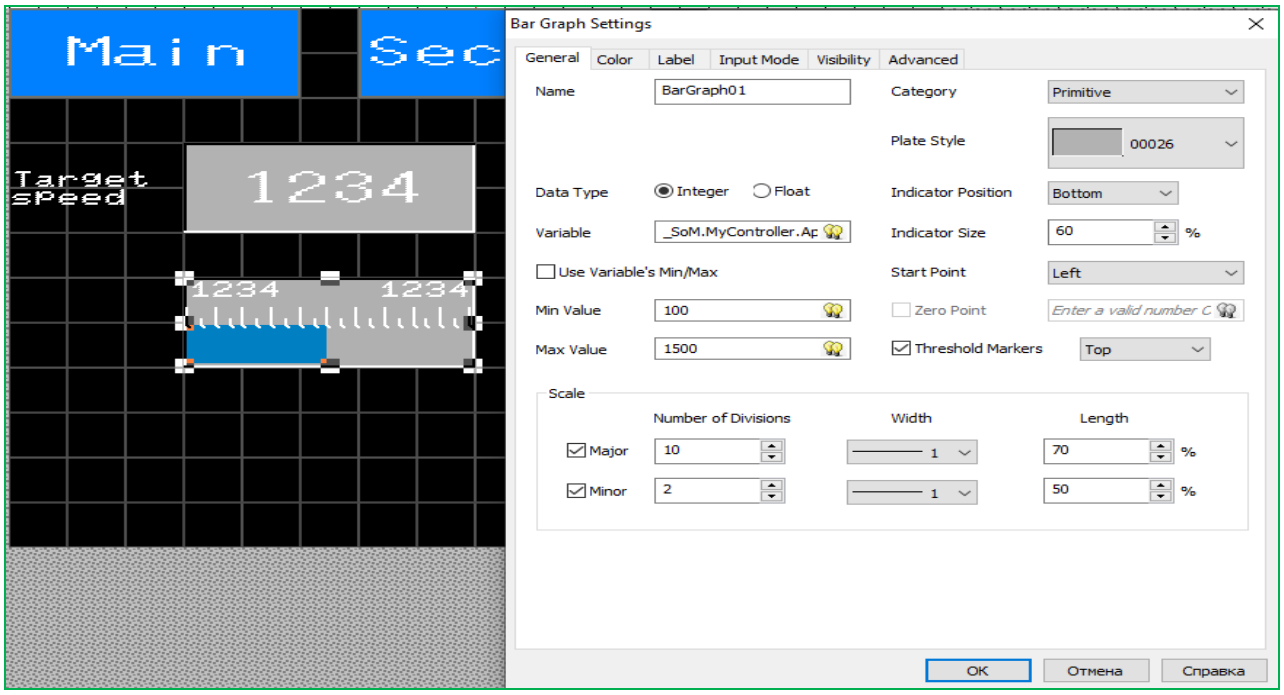

Figure 27. Bar Graph slider settings: General tab

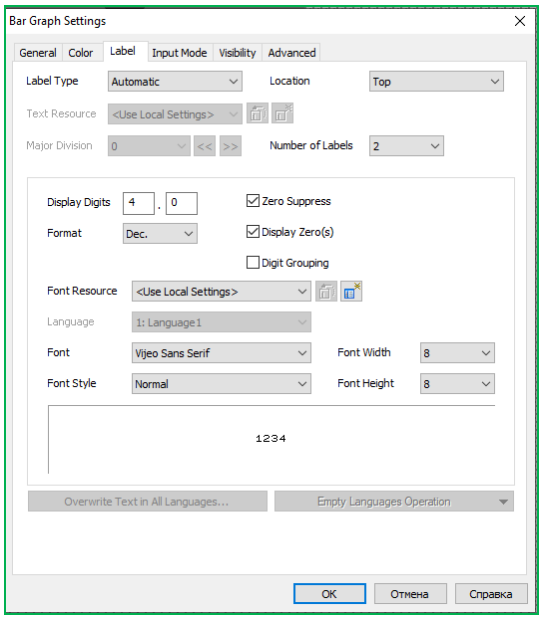

Figure 28. Bar Graph slider settings: Label tab

Above the “Power” button, create a label with the text “ATV930 Ethernet”.

Create a button to move the cargo up. For Operation parameter select Bit with Momentary ON behavior, for Lamp and Destination parameters select variable xCmdFwd_ATV930.

Create a button to move the cargo down. For Operation parameter select Bit with Momentary ON behavior, for Lamp and Destination parameters select variable xCmdRev_ATV930.

The appearance of the fully completed PanelSecond form is shown in Fig. 30.

CART DRIVE PAGE INTERFACE

Open the PanelSecond for secondary hoist actuator page form. Select all elements of the form, except for the top three page switching buttons. Copy the selected items to the clipboard. Open the PanelCart cart drive page form. Paste the previously copied elements onto the form.

Change the image of the "up" button to the image of the "left" arrow. For the Lamp and Destination parameters, select the variable xCmdRev_ATV930. Click the “Apply” button to accept the changes. In the Operation list, select Bit. For the new operation, select the behavior Momentary ON, for the Destination parameter, select the xLeft variable and click the “Add” button. This additional parameter is responsible for lighting the signal lamp on the stand when the cart moves to the left.

Change the image of the "down" button to the image of the "right" arrow. For the Lamp and Destination parameters, select the variable xCmdFwd_ATV930. Click the “Apply” button to accept the changes. In the Operation list, select Bit. For the new operation, select the behavior Momentary ON, for the Destination parameter, select the xRight variable and click the “Add” button. This additional parameter is responsible for lighting the signal lamp on the stand when the cart moves to the right.

For the options button, choose PanelOptions2 as the PanelID parameter. The appearance of the fully completed PanelCart form is shown in Fig. 31.

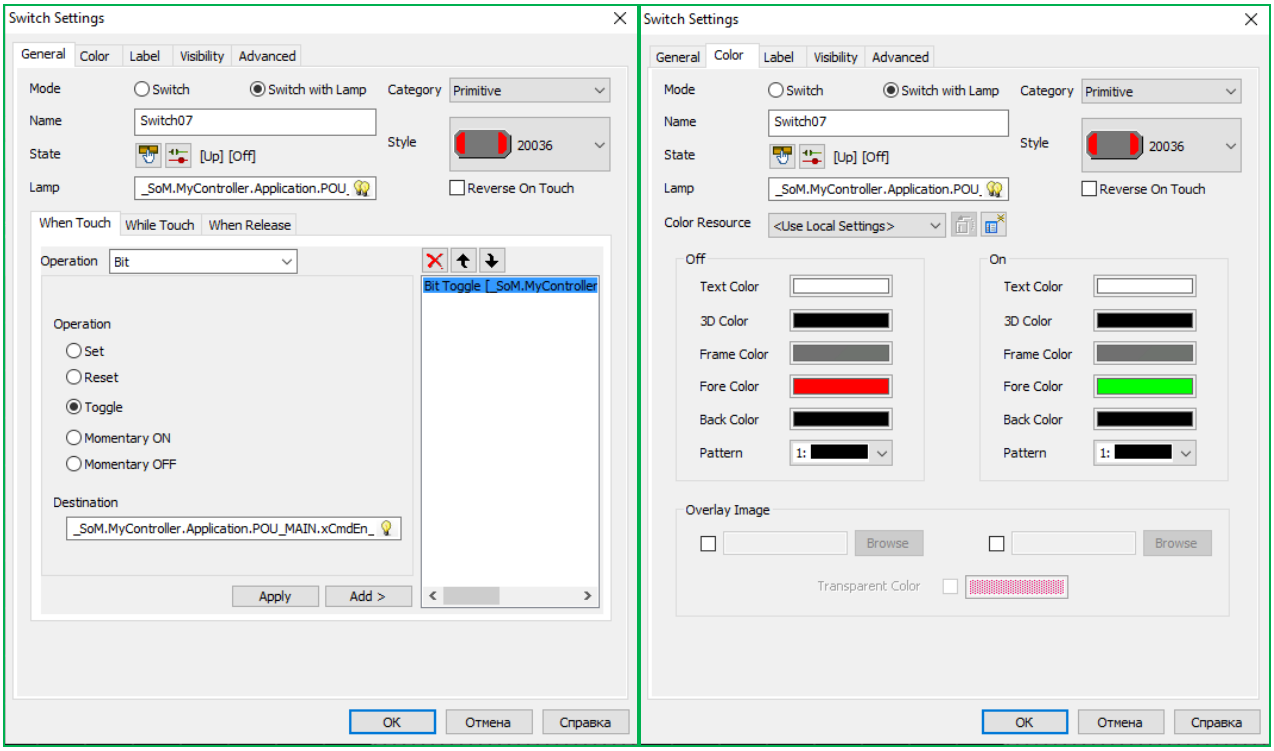

Figure 29. Setting the "Power" button of the PanelSecond page

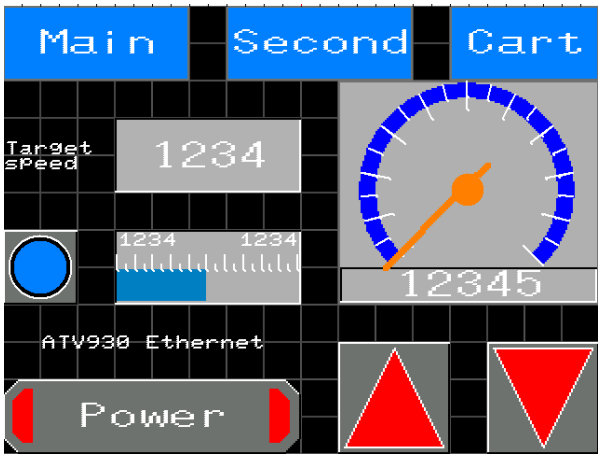

Figure 30. Auxiliary Lift Actuator Motion Control Page-from ATV930

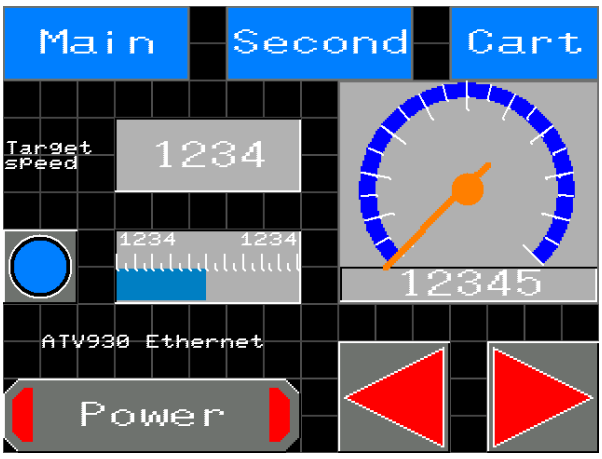

Figure 31. ATV930 Carriage Drive Motion Control Page

OPTIONS PAGE INTERFACE FOR ATV930

Open the PanelOptions1 drive options page form. Create an label on the form with the text "Acceleration, s (1…10)".

Next to the created label will place a Numeric Display element with the size of 2x2 cells. For Variable, select the variable xAcc1. For the Display Digits parameter, enter the number 2. On the Input Mode tab, put a checkmark next to the Enable Input Mode parameter. Activate the Overwrite Variable's Input Range option and set the Min Value field to 1 and the Max Value field to 10. Create a label on the form with the text "Deceleration, s (1…10)".

Next to the created label will place a Numeric Display element with the size of 2x2 cells. For the Variable parameter, select the variable xDec1. For the Display Digits parameter, enter the number 2. On the Input Mode tab, put a checkmark next to the Enable Input Mode parameter. Activate the Overwrite Variable's Input Range option and set the Min Value field to 1 and the Max Value field to 10. Create a label on the form with the text "Speed step, rev/min (1...10)".

Next to the created label will place a Numeric Display element with the size of 2x2 cells. For Variable, select the xStep1 variable. For the Display Digits parameter, enter the number 2. On the Input Mode tab, put a checkmark next to the Enable Input Mode parameter. Activate the Overwrite Variable's Input Range option and set the Min Value field to 1 and the Max Value field to 10.

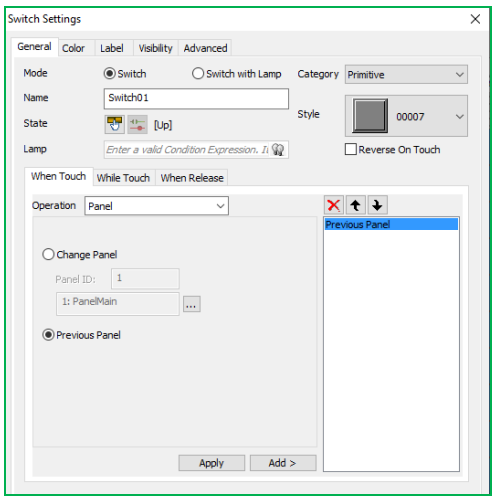

Figure 32. Setting the button to return to the previous page

Change the background color of the page to blue (Back Color parameter of the Property Inspector window). Create a button with the text "OK". The button setting is shown in Fig. 32. The appearance of the fully completed PanelOptions form is shown in Fig. 33.

Select all the created options page elements and copy them to the clipboard. Go to the PanelOptions2 form and paste the previously copied elements from the clipboard. Save the changes. This completes the creation of the touch panel interface project.

SIMULATION OF PANEL OPERATION

The simulation allows to check the operation of the project without actually loading it into real devices. During the simulation, one can monitor the status and values of all project variables.

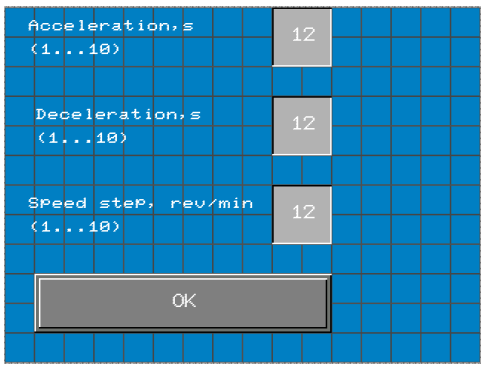

Figure 33. ATV930 Drive Options Page

First, it is need to simulate the controller. To do this, select the “Simulation” item in the “Online” main menu item in the SoMachine program. Then, in the same menu, select the "Login" item (choose a positive answer to all additional requests). To start the simulation, press the “F5” button on the keyboard.

Go to Vijeo Designer. Select “Simulation” from the “Build” menu item. At the same time, a touch panel window will appear, in which it is possible to check the operations of pressing buttons and initializing parameters. At the same time, the reaction to the performed operations will be visible in the SoMachine program (Fig. 34).

The state of the variables can be evaluated in the following windows:

- GVL – global variables;
- POU_Main, POU_LIMITS, POU_SPEED – programs;
- DI – input ports of the controller;
- DQ – output ports of the controller.

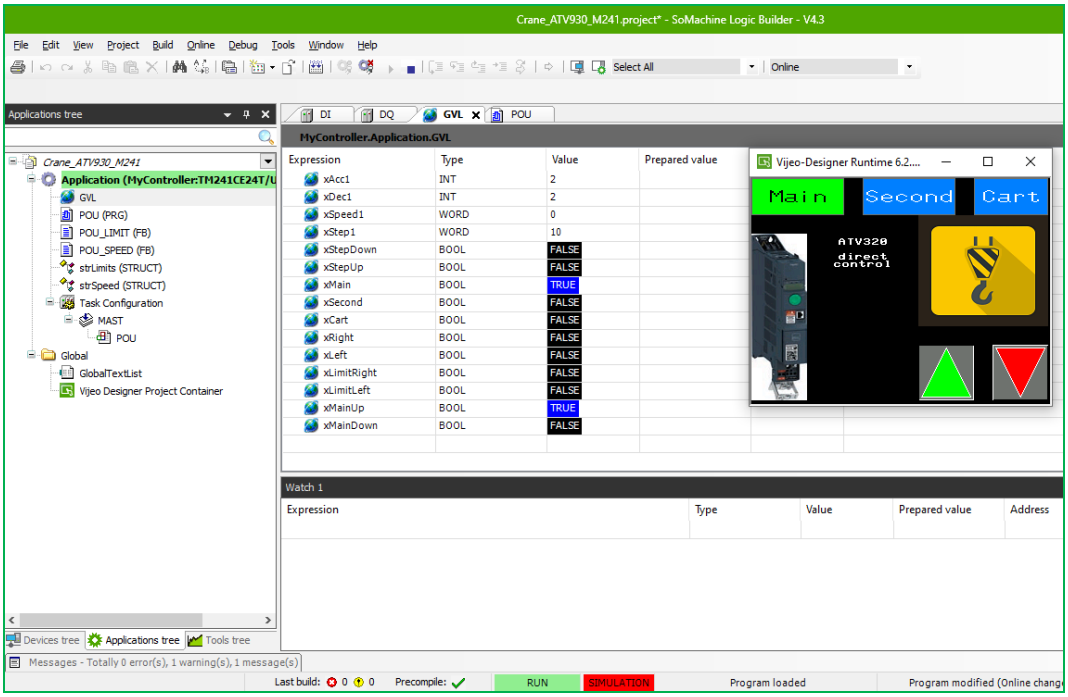

Figure 34. Project simulation in SoMachine and Vijeo Designer

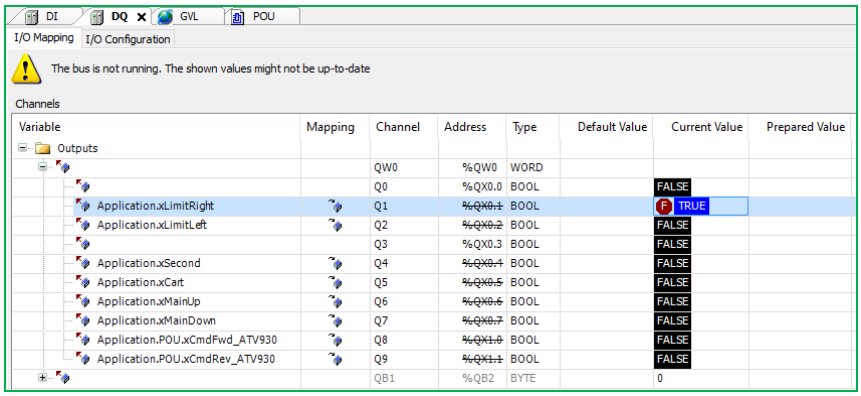

Figure 35. Manual assignment of the variable value during simulation

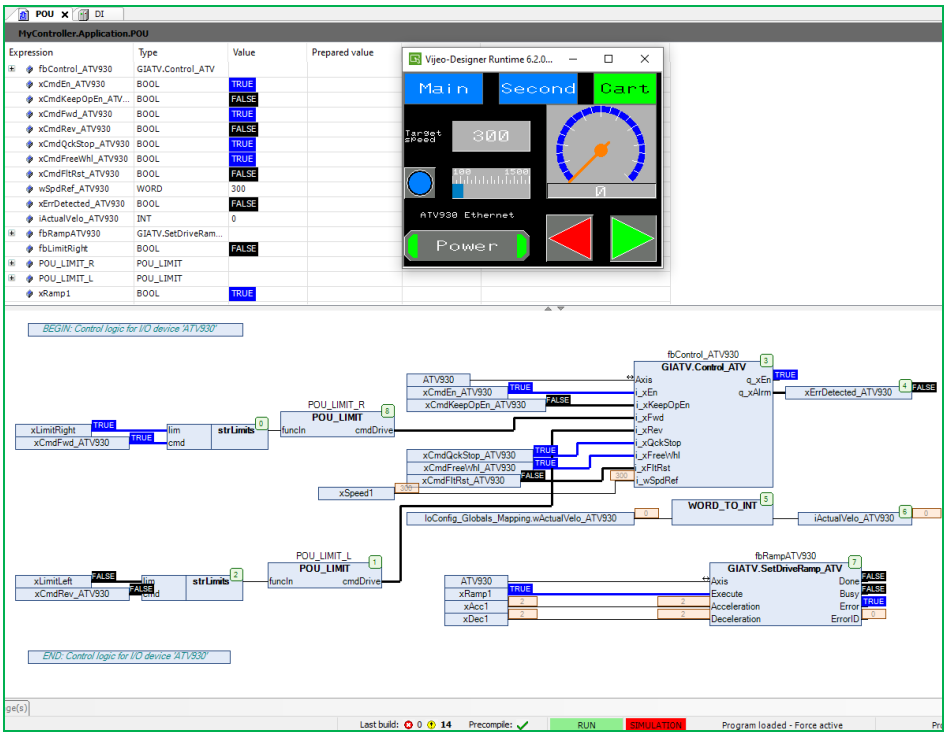

Figure 36. Simulation of limit switch actuation

During the simulation, it is possible change the value of any variable. For example, in the project, it is impossible to evaluate the operation of the input signals of the controller, since they are activated by the external circuit of the limit switches. However, one can do it manually. To do this, go to the DI (Digital Inputs) page.

In the table of controller inputs, in the cell next to the xLimitRight variable and double click on the "Prepared Value" column. The value "TRUE" will appear in the cell. Then press the "F7" key on the keyboard to accept the changes.

As a result, the variable xLimitRight will change its value from FALSE to TRUE. The result of such an operation is shown in Fig. 35.

Now go to the POU_Main program window. In the panel, go to the "Cart" tab and press the "right" button. In a correctly created project, the xCmdFwd_ATV930 input signal will change its value to TRUE, but the output of cmdDrive (the POU_LIMIT block) will have the value FALSE, which also corresponds to the black line (the black color of the lines in the simulation corresponds to the circuit without current, and the blue one to the circuit with current), which connects the cmdDrive port with the i_xFwd port of the GIATV.Control_ATV block (Fig. 36).

In real work, this will mean blocking the carriage movement command when the limit switch is closed. Check the correct operation of other controls, indicators and speed regulators. Check the status of the output ports of the controller when activating the buttons on the panel. At this stage, the simulation can be considered complete.

CONCLUSIONS

In the paper, the tasks of design, configuration and connection of the interface elements of the human-machine interface panel were successfully completed. Two types of control were implemented: with the ATV930 frequency converter via the Ethernet/IP protocol (or Modbus) and an intermediary in the form of the code and structure of the control system in SoMachine, as well as through the direct communication of commands from the HMISTU touch panel with the analog ports of the M241 controller to control the frequency converter ATV320.

All developments are original and do not repeat previous research. When setting up, remote activation/deactivation of the frequency converter is provided, as well as control of two overhead crane mechanisms (crane secondary hoist and carriage movement) from one frequency converter. Tested the operation of the panel in simultaneous simulation in the SoMachine controller configuration program.

The developed project was implemented in the created laboratory installation of a overhead crane with a control panel and is used for conducting research work of students, graduate students and scientists.

DISCLOSURE STATEMENT

No potential conflict of interests was reported by the author(s).

REFERENCES

1. Hirankitti, V. (2023). An object-oriented agent framework for SCADA. In A.K. Nagar, D. Singh Jat, D.K. Mishra, A. Joshi (eds.) *Intelligent Sustainable Systems*. LNNS, vol. 579 (pp. 687–696). Springer. https://doi.org/10.1007/978-981-19-7663-6_65

2. Hirankitti, V. (2019). An intelligent agent framework for SCADA. In *2019 First International Conference on Digital Data Processing (DDP)* (pp. 76–81). IEEE. https://doi.org/10.1109/DDP.2019.00024

3. Popescu, V.F., Gânsac, V., Comşa, O., Pârvu, M., & Pistol, S. (2022). Supervisory Control and data acquisition (SCADA) traffic simulation for controlling industrial processes and infrastructures. *Land Forces Academy Review, 27*(4), 423–440. https://doi.org/10.2478/raft-2022-0051

4. Kibulungu, J.W., Laseinde, O.T. (2023). Automatic control system based on Industry 4.0, PLC, and SCADA. In A.K. Nagar, D. Singh Jat, D.K. Mishra, A. Joshi (eds.) *Intelligent Sustainable Systems*. LNNS, vol. 578 (pp. 183–197). Springer. https://doi.org/10.1007/978-981-19-7660-5_16

5. Kiangala, K.S., & Wang, Z. (2019). An Industry 4.0 approach to develop auto parameter configuration of a bottling process in a small to medium scale industry using PLC and SCADA. *Procedia Manufacturing, 35*, 725–730. https://doi.org/10.1016/j.promfg.2019.06.015

6. Merchan, D.F., Peralta, J.A., Vazquez-Rodas, A., Minchala, L.I., & Astudillo-Salinas, D. (2017). Open source SCADA system for advanced monitoring of industrial processes. In *2017 International Conference on Information Systems and Computer Science (INCISCOS)* (pp. 160–165). IEEE. https://doi.org/10.1109/INCISCOS.2017.9

7. Fitriani, S., & Sofyan, Y. (2020). Simulator human machine interface (HMI) using visual basic on the SCADA system. *IOP Conference Series: Materials Science and Engineering, 830*(3), 032016. https://doi.org/10.1088/1757-899X/830/3/032016

8. Almas, M.S., Vanfretti, L., Løvlund, S., & Gjerde, J.O. (2014). Open source SCADA implementation and PMU integration for power system monitoring and control applications. In *2014 IEEE PES General Meeting | Conference & Exposition* (pp. 1–5). IEEE. https://doi.org/10.1109/PESGM.2014.6938840

9. Endi, M., & Elhalwagy, Y. (2010). Three-layer PLC/SCADA system architecture in process automation and data monitoring. In *2010 the 2nd international conference on computer and automation engineering* (Vol. 2, pp. 774–779). IEEE. https://doi.org/10.1109/ICCAE.2010.5451799

10. Padhee, S., & Singh, Y. (2011). Data logging and supervisory control of process using LabVIEW. In *IEEE Technology Students' Symposium* (pp. 329–334). IEEE. https://doi.org/10.1109/TECHSYM.2011.5783838

11. Hafeez, S.M., & Ghauri, Y.J. (2014). Supervisory control and FPGA based DAQ card. *Applied Mechanics and Materials, 666*, 64–68. https://doi.org/10.4028/www.scientific.net/AMM.666.64

12. Kumar, N., & Kumar, U. (2012). Simulation of virtual SCADA system using LabVIEW. In *2012 IEEE 5th India International Conference on Power Electronics* (pp. 1–5). IEEE. https://doi.org/10.1109/IICPE.2012.6450456

13. Sita, I.V. (2012). Building control, monitoring, safety and security using collaborative systems. In *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems* (pp. 662–667). IEEE. https://doi.org/10.1109/iNCoS.2012.107

14. Marosan, A.I., Constantin, G., Barsan, A., Crenganis, M., & Girjob, C. (2020). Creating an ethernet communication between a Simatic S7-1200 PLC and Arduino Mega for an omnidirectional mobile platform and industrial equipment. *IOP Conference Series: Materials Science and Engineering, 968*, 012022). https://doi.org/10.1088/1757-899X/968/1/012022

15. Electric, S. (2014). Schneider Electric: Estratégia SCADA para os Próximos Três Anos. *Neutro à Terra*, (14), 61–64. https://doi.org/10.26537/neutroaterra.v0i14.420

16. Fandidarma, B., Sunaryantiningsih, I., & Pratama, A. (2022). Pengatur suhu ruangan tertutup menggunakan PLC Schneider TWIDO COMPACT berbasis SCADA-WONDERWARE INTOUCH. *ELECTRA: Electrical Engineering Articles, 2*(2), 01-11. https://doi.org/10.25273/electra.v2i2.12246

17. Yang, B., Qiao, L., Zhu, Z., & Wulan, M. (2016). A metamodel for the manufacturing process information modeling. *Procedia CIRP, 56*, 332-337. https://doi.org/10.1016/j.procir.2016.10.032

18. Haba, S.A., Oancea, G. (2023). Study on design and manufacturing of an engine block using digital tools. In P. Kyratsis, A. Manavis, J.P. Davim (eds.) *Computational Design and Digital Manufacturing*. MINEN (pp. 167–203). Springer. https://doi.org/10.1007/978-3-031-21167-6_9

19. Querejeta, M.U., Etxeberria, L., & Sagardui, G. (2021). Virtual commissioning in machine tool manufacturing: a survey from industry. *DYNA, 96,* 612–619. https://doi.org/10.6036/10244

20. Lechler, T., Fischer, E., Metzner, M., Mayr, A., & Franke, J. (2019). Virtual commissioning – Scientific review and exploratory use cases in advanced production systems. *Procedia CIRP, 81*, 1125–1130. https://doi.org/10.1016/j.procir.2019.03.278

21. Aksonov, Y., Tsegelnyk, Y., Piddubnyi, O., Kombarov, V., Plankovskyy, S., & Piddubna, L. (2022). Interactive design of CNC equipment operator panels. *International Journal of Mechatronics and Applied Mechanics, 2022*(12), 156–162. https://doi.org/10.17683/ijomam/issue12.23

22. Zhang, P. (2010). Human-machine interfaces. In *Advanced Industrial Control Technology* (pp. 527–555). William Andrew. https://doi.org/10.1016/B978-1-4377-7807-6.10013-0

23. Gong, C. (2009). Human-machine interface: Design principles of visual information in human-machine interface design. In *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics* (Vol. 2, pp. 262–265). IEEE. https://doi.org/10.1109/IHMSC.2009.189

24. Massaro, A. (2021). Human–machine interfaces. In *Electronics in Advanced Research Industries*. John Wiley & Sons. https://doi.org/10.1002/9781119716907.ch3

25. Schneider Electric. (2014). Vijeo Designer Tutorial.

26. Sărăcin, C.G., Deaconu, I.D., & Chirilă, A.I. (2019). Educational platform dedicated to the study of programmable logic controllers and the human-machine interface. In *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)* (pp. 1–4). IEEE. https://doi.org/10.1109/ATEE.2019.8725021

Розробка SCADA для електричного приводу у Vijeo Designer (Schneider Electric)

Олександр Аксьонов, Владислав Плюгін, Євген Цегельник, Олексій Словіковський, Олексій Дунєв, Андрій Єгоров

Анотація. Стаття присвячена розкриттю побудови системи SCADA для управління режимами роботи і моніторингом електричного приводу мостового підйомного крану. Матеріал наданий у вигляді покрокової інструкції з побудови такої системи у програмі Vijeo Designer. Детально показано створення нового проєкту, елементів інтерфейсу, прив'язка змінних з елементами інтерфейсу, налаштування поведінки динамічних елементів інтерфейсу. Розкрито питання зв'язку глобальних змінних візуальних елементів інтерфейсу з контролером M241 та програмою у SoMachine. Розроблений інтерфейс дозволяє керувати трьома механізмами мостового крану: головний підйом, допоміжний підйом та пересування візка. Показано, яким чином виконати управління перетворювачів частоти, які зв'язані з контролером через різні типи зв'язку: прямий аналоговий з контролером, та через протокол Ethernet/IP. Робота буде корисна для академічних цілей у навчанні студентів, дослідників, а також для інженерів промислових підприємств.

Ключові слова: пристрій людського інтерфейсу, SCADA, електропривод, Vijeo Designer, пульт управління, система управління, дистанційне керування

NOTES ON CONTRIBUTORS

Oleksandr Aksonov
oleksandr.aksonov@kname.edu.ua
Postgraduate Student
Department of Urban Power Supply Systems and Power Consumption
O. M. Beketov National University of Urban Economy in Kharkiv, Kharkiv, Ukraine
https://orcid.org/0000-0002-1870-524X
https://scopus.com/authid/detail.uri?authorId=57809116200

Vladyslav Pliuhin
vladyslav.pliuhin@kname.edu.ua
D.Sc., Professor
Department of Urban Power Supply and Consumption Systems
O. M. Beketov National University of Urban Economy in Kharkiv, Kharkiv, Ukraine
https://orcid.org/0000-0003-4056-9771
https://www.webofscience.com/wos/author/record/F-4627-2018/
https://scopus.com/authid/detail.uri?authorId=57204286328

Yevgen Tsegelnyk
y.tsegelnyk@kname.edu.ua
Ph.D., Senior Researcher
Department of Automation and Computer-Integrated Technologies
O. M. Beketov National University of Urban Economy in Kharkiv, Kharkiv, Ukraine
https://orcid.org/0000-0003-1261-9890
https://www.webofscience.com/wos/author/record/J-1570-2015/
https://scopus.com/authid/detail.uri?authorId=57192961558

Oleksii Slovikovskyi
slovikovskyi@nubip.edu.ua
Postgraduate Student
Department of Automation and Robotic Systems named by I. Martynenko
National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine
https://orcid.org/0000-0001-8912-6256

Oleksii Duniev
oleksii.duniev@khpi.edu.ua
Ph.D., Associate Professor
Department of Electrical Machines
National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
https://orcid.org/0000-0001-8196-7077
https://www.webofscience.com/wos/author/record/G-3310-2016/
https://scopus.com/authid/detail.uri?authorId=57497865000

Andrii Yehorov
andrii.yehorov@khpi.edu.ua
Ph.D., Associate Professor
Department of Electrical Machines
National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
https://orcid.org/0000-0001-6643-4256
https://www.webofscience.com/wos/author/record/G-3960-2016/
https://scopus.com/authid/detail.uri?authorId=57200639200