

Development of Software for the Implementation of Automated Reserve Input Modes Operation

Bohdan Orobchuk[†], Oleh Buniak[†], Ivan Sysak^{*,†}, Serhii Babiuk^{*,†}, Ihor Bodnarchuk[†], Vadym Koval[†]

Ternopil Ivan Puluj National Technical University, Ruska str. 56, Ternopil, 46001, Ukraine

Abstract

The problem of the development of automated reserve input system and software for communication with OMD switch controller produced by ABB company based on the laboratory stand, used for laboratory works aimed at the research and control of power system operation modes of electric power systems at the Department of Electrical Engineering of Ternopil Ivan Puluj National Technical University is considered in this paper. Software based on Modbus RTU protocol has been developed for data exchange between the laboratory stand and OMD device. The development and implementation of the software is carried out in the Microsoft Visual Studio environment. The creation of diagrams is performed in the Borland Together design environment. The project schedule is implemented by means of Microsoft Project, and the operation of communication and Modbus registers is implemented due to the Modbus Poll program. The program makes it possible to evaluate the state of OMD device and to analyze the operating modes of the reserve input system elements: selection of the source for transferring electricity to the load between the main and secondary sources, the status of the connected electrical network and switching devices. The animation mode has also been developed to present the functions of the OMD device without physical connection to the device.

Keywords

automated reserve input, automatic control, laboratory stand, load switch, microprocessor control system, software

1. Introduction

Modern technological systems operate using electricity, which is taken for granted. Electricity is essential in industry, transportation, and everyday life. The common requirement of all consumers is the continuity of power supply or its rapid restoration in

CITY'2024: 2nd International Workshop on Computer Information Technologies in Industry 4.0, June 12–14, 2024, Ternopil, Ukraine

*Corresponding author.

[†]These authors contributed equally.

✉orobchuk_b@tntu.edu.ua (B. Orobchuk); buniak0203@gmail.com (O.Buniak); sisak.tntu@gmail.com (I. Sysak); sermuk@gmail.com (S.Babiuk); bodnarchuk.io@gmail.com (I. Bodnarchuk); koval_vadym@tntu.edu.ua (V. Koval)

ORCID: 0000-0002-6375-2440 (B. Orobchuk); 0000-0001-9304-8254 (O.Buniak); 0000-0002-2315-7911 (I. Sysak); 0000-0001-5318-8586 (S.Babiuk); 0000-0003-1443-8102 (I. Bodnarchuk); 0000-0002-7427-6507 (V. Koval)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

case of interruptions. Although interruptions in electricity supply to household consumers are of emergency nature, they do not cause significant damage. Instead, the peculiarities of the operation of most technological and technical systems in industry and transport require uninterrupted power supply or, in case of the problem, its restoration as soon as possible [1].

The generation of electrical energy is carried out by various methods, but in the systems of distribution by electrical networks, the equipment that can recognize faults in the system and replace the electrical network connected to the load with another one is installed in order to ensure uninterrupted supply of electrical energy. In industry, the backup power plant or own generator which should be put into operation in the event of power outage are used as the second source of electricity at electricity distribution units. In some cases, the selection of backup power supply link is carried out in manual mode due to electrical switches for direct and reverse action significantly slowing down the restoration of power supply to consumers in case of failure [2].

Taking into account these problems, ABB company produced a series of OMD products that can be connected to electrical network switches and provide automatic switching function. OMD device constantly checks the operating conditions of two power sources; and if failure occurs in the power network that has the priority of the main power source, the backup power source is connected to the load. With the specified automated switching of power supply links, OMD device ensures significantly shorter interruption time in power supply of technological and technical equipment, and the switching procedure itself does not require manual mode. Some of OMD series devices have generator control functions and can automatically start or stop the generator. The proposed technical system uses OMD800 device, which is more universal in terms of extended package of functions, it can provide status information about the current state of the switch and connected power network links via Modbus protocol, for example, to the dispatcher of the controlled power network link [3].

The automated system based on the laboratory stand and software, which was implemented in the educational process of the Department of Electrical Engineering at Ternopil Ivan Puluj National Technical University for acquaintance and application of OMD800 device functions is considered in this paper.

The device can be demonstrated without being connected to the computer. Instead, when the device is connected to the computer, the software analyzes the status of the OMD800 device using the Modbus protocol. The objective of the developed automatic system based on the laboratory stand and software is to visualize the functioning of devices of OMD product series during laboratory work in the laboratory of "Automated Dispatch Control Systems" in "Power Supply Management Systems" educational course. The students will have the opportunity to study the operation of the switch controlled by OMD device, get acquainted with the processes of data exchange during switching between the main and backup power supply links, and carry out the process of generator control [4].

2. Development of the laboratory stand and connection to the simulator

The main task in the construction of the automated system was to ensure simplicity in conducting the laboratory experiment by students [5]. Graphical representation of equipment operation using diagrams made it possible to perform the software implementation of data exchange as clear and informative as possible. By default, the working language for constructing the software package is English. For more flexible application in the educational process, there is an option for language selection (English, Ukrainian) in the program texts displaying.

Figure 1 shows the main devices of the laboratory stand (without computer): 1 - OMD800 automatic control unit; 2 - load reversing switch controlled by OMD device; 3 - load reversing switch drive; 4 - converter of ADAM 4520 RS-232 into RS-485; 5 - ADAM converter power supply (Siemens); 6 - power electrical panel (Figure 2).

In order to increase visualization and practical application, the developed system was integrated into the training SCADA-system "Energy" of the hardware - software complex "Arrow" [6], which is designed as a simulator of the automated dispatch control system for research and control of the operating modes of electric power systems [7-9].

To exchange data between hardware - software complex "Arrow" and OMD800 device, the program based on the Modbus RTU protocol was developed [10]. Only a part of the available commands of OMD800 device register, which in case of need can be added to the laboratory stand software are used in the program block presented below.

The communication channel RS232-RS485 is used to exchange data with the devices of the laboratory stand. The data are transferred into the equipment model, where the messages are processed. Next, the data are transferred to the real model, where the final processing is carried out: fixing the message in the database, analysis of emergency situations, , and saving information about the current status of the model. During the work, the student uses the graphical model that provides a request for input of the real model and outputs it in a form familiar to the user - mnemonic schemes (рис. 3).



Figure1: Laboratory stand equipment



Figure2: Power electrical panel for the laboratory stand power supply:
 1 - automatic switch for personal computer powering; 2 - automatic switch for power supply of the load reversing switch drive power supply; 3 - automatic switch of three-phase 380V power line supplied to the load reversing switch; 4 - automatic switch of three-phase 380V generator supplied to the load reversing switch.

Modbus RTU device card file

[GENERAL]

AddressDecrement=0

[REQUEST]

; Start address, number of registers, Modbus command, group 101, whether included in the general protocol

request01=58,14,3,1,1

request02=150,15,3,1,1

request03=166,11,3,1,1

request04=250,3,3,1,1

[TVLIST]

; Start address, offset (for a bit), data type, scale (coefficient), start of scale

; Data types 0: ui16 1: i16 2: ui32 3: i32 4: float32

; 5: float48 6: Int64 7: float64 8: float80

; Start address, offset (for bit), data type, scale

1_U1 - phase A voltage of the first line (with an accuracy of 0.1 V)

tv001=150,0,1,0.1

;1_U2 - phase B voltage of the first line (with an accuracy of 0.1 V)

tv002=152,0,1,0.1

;1_U3 - phase C voltage of the first line (with an accuracy of 0.1 V)

tv003=154,0,1,0.1

;1_U12 - voltage between phases AB of the first line (with an accuracy of 0.1 V)

tv004=158,0,1,0.1

;1_U23 - voltage between phases BC of the first line (with an accuracy of 0.1 V)

tv005=160,0,1,0.1

;1_U31 - voltage between phases CA of the first line (with an accuracy of 0.1 V)

tv006=162,0,1,0.1

;2_U1 - phase A voltage of the second line (with an accuracy of 0.1 V)

tv007=164,0,1,0.1

;2_U2 - phase B voltage of the second line (with an accuracy of 0.1 V)

tv008=166,0,1,0.1

;2_U3 - phase C voltage of the second line (with an accuracy of 0.1 V)

tv009=168,0,1,0.1

;2_U12 - voltage between phases AB of the second line (with an accuracy of 0.1 V)

tv010=172,0,1,0.1

;2_U23 - voltage between phases BC of the second line (with an accuracy of 0.1 V)

tv011=174,0,1,0.1

;2_U31 - voltage between phases CA of the first line (with an accuracy of 0.1 V)

tv012=176,0,1,0.1

;1_F - frequency of the first power line (with an accuracy of 0.1 V)

tv013=250,0,1,0.1

;2_F - frequency of the second power line (with an accuracy of 0.1 V)

tv014=252,0,1,0.1

;Start address, offset (for a bit)

;Reg_SL_Status - Register of additional load status

```

tv015=60,0,0,1
;Reg_Generator_alarm - Generator failure signal register
tv016=61,0,0,1
;Reg_Force_manual - Register for forced switching to manual control mode
tv017=62,0,0,1
;Reg_Force_Comutation - Register of forced comutation
tv018=63,0,0,1
;Reg_Generator_start - Generator start register
tv019=64,0,0,1
;Reg_Inhibit_switching - Switching inhibit register
tv020=65,0,0,1
;Reg_Inhibit_remote - Remote control inhibit register
tv021=66,0,0,1
;Reg_Remote_0 - Register for remote switching to position I
tv022=67,0,0,1
;Reg_Remote_1 - Register for remote switching to position I
tv023=68,15
;Reg_Remote_2 - Register for remote switching to position II
tv024=69,15
;Reg_Man_Back_Switching - Reverse switching register in manual mode
tv025=70,15
;Reg_Emergency_Stop - Emergency stop register
tv026=71,15
;Reg1_status - No-voltage register
tv027=58,0,0,1
;Reg2_status - Under-voltage register
tv028=59,0,0,1
[OK]
ok=ok

```

The developed program file of the Modbus RTU device map and OMD800 automatic control unit itself are the real model of the training SCADA system "Energy". To integrate this model into the SCADA system environment, the graphical model (Figure 3) and configuration file containing the following data were created.

Configuration file

```

[GENERAL]
ObjectCount=1 - number of substations
ChannelsCount=1 - number of communication channels
DisableCrashLog=1 — disable the automatic creation of logs (by default, log data is
written to the Logs directory or the application startup directory)
CheckInterval=1000 - control checkup period in msec.
CommonRequestInterval=30000 — common request interval in msec.

```

```
; Modbus
[Channel1] - 1 communication channel is used
Type=1 - channel type (2 - TCP-IP, 1 - RS232)
Thread=1 - thread number
Port=7 - port number
Prm=19200-8-N-1 -RS232 connection parameters (19200 - frequency, 8 - 8 bits, N
- parity, 1 - stop bit)
ByteTimeOut=200 - waiting time for the next byte
DataReadTimeOut=3500
DeffaultTimeout=2000
Description=Modbus
[Object1]
Type=7
ModbusTCP=0 - The protocol used is not the classic Modbus TCP, but a specialised
Modbus RTU protocol is used
Address=42 - model address
Channel=1 - 1st communication channel
Paused=0 - survey delay time
// Modbus RTU map file name of the device
Map=ABB_OD800 - object model
Description=AVR(ABB) - model description
[MainForm] - parameters of the main window for viewing data exchange logs (filled
in automatically).
Top=179
Left=738
Width=890
Height=627
LVHeight=150
[ModelState] - parameters of the window for graphic diagram placement of the
model
Top=17
Left=247
Width=887
Height=974
[LogForm0]
Top=182
Left=182
Width=698
Height=316
[LogForm1]
Top=500
Left=32
Width=698
Height=316
```

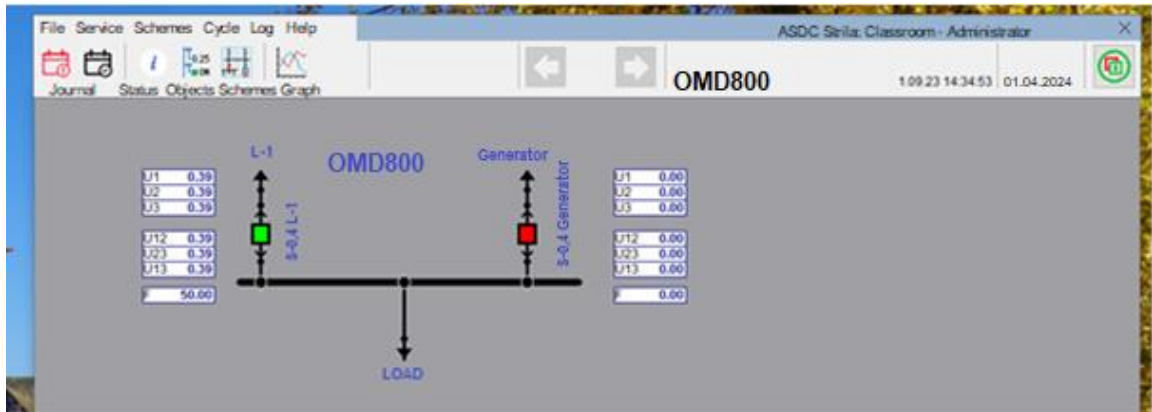


Figure 3: Window of the graphical model of communication between the laboratory simulator and OMD800

3. Software development for the laboratory stand

The development and implementation of the software was carried out in Microsoft VisualStudio environment, where comprehensive programming language tools such as VisualBasic, Visual J#, Visual C++ and Visual C# were available [11].

The diagrams were created in the BorlandTogether design environment, the project schedule was implemented due to Microsoft Project, and the work of communication and Modbus registers was performed using ModbusPoll program [12].

The basic SwitchEnergy program consists of several modules, each of which performs its own task.

The communication module contains ModbusConnector class with subprograms for Modbus communication, in particular, searching for free COM ports, opening and closing the connection, creating a message, sending, receiving and checking errors.

Forms module contains all the forms that are used in the program, the main one being MainForm. When the program starts, it also creates. While launching the program, ModbusDataForm, which contains verbal explanation of the values of OMD device registers is created. Depending on the method the program is used, DelayForm that sets the delay modes at different points of the electrical connection (animation mode) or ModbusStatusForm (real device operating mode) containing verbal explanation of the device - the current status are created.

Graphics module contains classes that process program graphics.

Languages module contains classes for the program languages. The division of the program into categories and modules is shown in Figure 4.

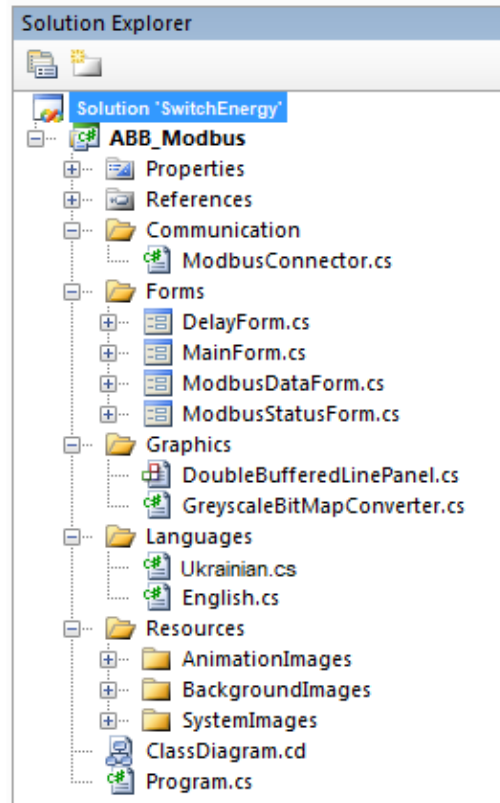


Figure4: Program division into classes and modules

3.1. Implementation of communication with Modbus

Modbus request is created in ModbusConnector class, which contains subprograms related to establishing and maintaining connection. When the user selects the correct device mode to be used in the program, the process where COM communication ports found on the computer are determined by the operating system is started. One Modbus message is sent to each of the available waiting for response ports. If the response is received, the operating port is found, and from this moment this port is used for Modbus communication. Modbus message itself, which requests certain registers from a certain Modbus device, is created using the following GenerateModbusQuery subprogram:

```
private void GenerateModbusQuery
(ushort registers, ref byte[] message)
{
    // Address of the device is 1
    byte address = 1;
    // Starting register is always 2000
    ushort start = 2000;
    // Array to receive CRC bytes
    byte[] CRC = new byte[2];
    // Modbus query is created here. message[1] is funct. code
    message[0] = address;
```

```

message[1] = 3;
message[2] = (byte)(start >> 8);
message[3] = (byte)start;
message[4] = (byte)(registers >> 8);
message[5] = (byte)registers;
// After query created, calculate it's CRC value
CalculateCRCvalues(message, ref CRC);
message[message.Length - 2] = CRC[0];
message[message.Length - 1] = CRC[1];
}.

```

Since SwitchEnergy program is supposed to connect only to one Modbus device and the data are read from predefined registers, these initial conditions are already determined as fixed in the code. Modbus request is addressed to the slave device with identifier equal to 1 and uses the function code number 3, which indicates the number of registers [13]. Eight registers are read from the device, starting with register 2000 [14]. Modbus message is created from this information,. The request is sent to the functional communication port determined from the list of COM ports previously received from the operating system due to SendQuery subprogram:

```

try
    {
        // Keep track of sent messages
        Sent Query++;
        // Write message to serial port
        30
        Modbus SerialPort. Write(message, 0, message.Length);
        // Get the response
        Get Response(ref response);
    }
catch (Exception)
    {
        // If error received, set communication error to true
        Communication Error = true;
        Return false;
    }
}.

```

The number of both sent and received messages is stored. If these numbers differ from each other, the connection has failed and the user receives notification. In case of connection failure, wait for 10 seconds until the connection is restored. If the connection is restored within this time, Modbus requests will continue normal operation.

The response to the sent Modbus request is stored by the subprogram: GetResponse:

```

private void GetResponse(ref byte[] response)
    {
        // Get data from serial port
        for (int i = 0; i<response.Length; i++)

```

```
    {  
        // and save it to response -array  
        response[i] = (byte)(modbusSerialPort.ReadByte());  
    }  
    }.
```

The received response is read by the serial port into the byte table one byte at a time until the length of the received message fills in the table. The received message always has a standard size, regardless of its content, if the request type remains unchanged. After that, CRC checking is performed on the received message ensuring the integrity of the received packet. CRC checking of the received message is verified, and if it is correct, the message decoding continues.

3.2. Graphical representation of data

The power connection of the primary and secondary sources and the load are graphically represented in the program as well (Fig. 5). While using the program, the student observes: the control status of the generator, which power source is connected to the load, which electrical lines are energized. This information is transferred from Modbus bus and is used to update the chart to reflect the real status.

If animation is the operating mode of the program (the program is used without OMD device), then the chart graphic is created on the basis of Modbus bus data due to ModbusDataInjector subprogram. The software representation of the lines on the diagram is quite time-consuming during programming, in the real device mode, the animation on the lines is not displayed, and the connection of lines to the power supply (the color of the image representation) is obtained directly from Modbus request data (regardless of whether the primary or secondary power source is energized). Animation is not displayed in the operating mode of the real device, the student is able to observe the current situation and changes in electric network.

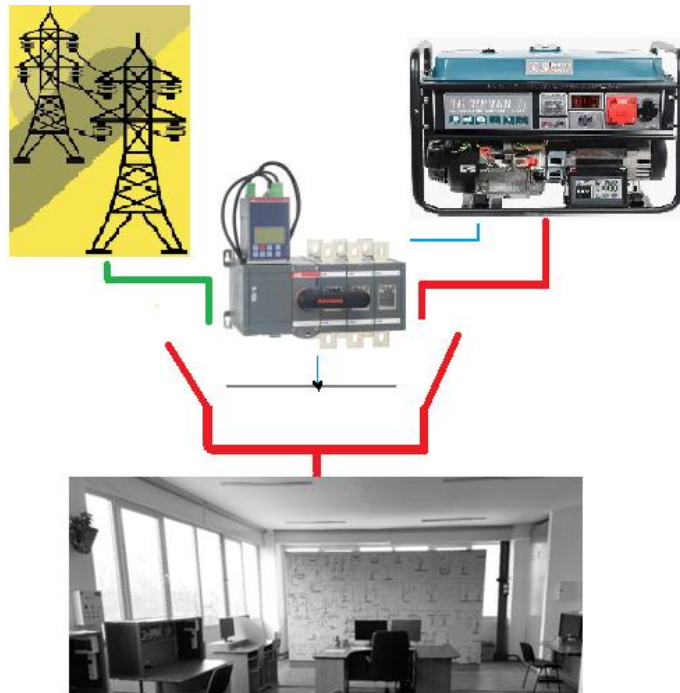


Figure 5: Graphical representation of the power supply to the equipment and the connection status. The primary network (top left) is generating electric power (green conductor), but it is not connected to the laboratory load (bottom, black and white).

The following program fragment describes the status of the system when the main power supply source is connected to the load and the generator is not operating. Graphic lines display information received from Modbus registers and the linear image requires panel updating. Since the Modbus values have changed after the previous register reading, ModbusStatusForm window is also updated in order to reflect the new system status:

```

if (modbusData[0]==0 &&modbusData[1]==1 &&modbusData[5]==1
&&modbusData[6] == 2)
{ // Devicestatus 1
  // LINE1:      Voltage OK  AND
  // LINE2:      No voltage  AND
  // Generator:   OFF       AND
  // ChangeoverSwitch: Inposition I
  // drawLineParams: parametersusedinline draw
  // [0] = isprimarynetworkproducingpower
drawLineParams[0] = true;
  // [1] = isbackupnetworkproducingpower
drawLineParams[1] = false;
  // [2] = isprimarynetwork SWITCHED
drawLineParams[2] = true;
  // [3] = isbackupnetwork SWITCHED

```

```

drawLineParams[3] = false;
    // [4] = is LOAD on
drawLineParams[4] = true;
    // Forcereshofthepanel
centerPanel.Refresh();
    // Booleanfordetermin. istheload ON or OFF. This is
    // usedinthemodbusstatusform, indrawingthe
    // colouredboxnexttothe text
systemIsRunning = true;
modbusStatusForm.Refresh();
    // Inthisstage, Gridisonandbackupis off
primaryPowerPictureBox.Image = gridOnImage;
backupPowerPictureBox.Image = backupOffImage;
    // Readtextfromstagesarray to modbusstatusform
modbusStatusFormText = stages[8];
modbusStatusFormStage = 8;
modbusStatusForm.ModbusStatusLabel.Text = stages[8];
systemIsRunning = true;
    // Ifmodbusstatusformisshowing, updatepreviousdata
if (modbusStatusFormShowing)
updateModbusStatusForm();
    // Drawloaddependingifthepowerisreallyon or not
if (modbusData[0] == 0 &&modbusData[5] == 1)
loadPictureBox.Image = loadOnImage;
elseif (modbusData[1] == 0 &&modbusData[5] == 3)
loadPictureBox.Image = loadOnImage;
else
loadPictureBox.Image = loadOffImage;
    // Drawcurrentpicturetopictureboxandshowit
Imageimage = Image.FromFile(animationImageDirPath +
    "01_PrimaryUp_SecondaryDown_PowerOn.gif");
centerPictureBox.Image = image;
}.

```

The images on the program's diagram inform the students whether the given object is powered or not, in colored or black - white image. If necessary, the student can change the image to any other color or make black and white copy of it. The image is processed to black and white colour by GreyscaleBitmapConverter class, keeping the necessary transparency:

```

for (currentPixelHeight = 0;
currentPixelHeight<convertedBitmap.Height;
currentPixelHeight++)
{

```

```

        for (currentPixelWidth = 0; currentPixelWidth <
            convertedBitmap.Width; currentPixelWidth++)
        {
            Color color = sourceBitmap.GetPixel
                (currentPixelWidth, currentPixelHeight);
            lumaValue = (int)
                (color.R * 0.3 + color.G * 0.59 + color.B * 0.11);
            convertedBitmap.SetPixel
                (currentPixelWidth, currentPixelHeight, Color.FromArgb
                (color.A, lumaValue, lumaValue, lumaValue));
        }
    }.

```

3.3. Organization of laboratory stand communications

The communication connection between OMD device and computer is created by means of required converters, the types of which depend on the computer configuration. If the computer has USB port, USB to RS232 and RS232 to RS485 converters are required. If the computer has RS232 port, only RS232 to RS485 converter is required. In this case, the communication connection is performed by means of twisted-pair cable to the connection on the top of OMD800 device [15]. Communication between OMD device and computer takes place according to Modbus protocol, where the main device sends the request to which the involved device responds specified in the request. Figure 6 shows a Schematic explanation of the operation of the presented automated system equipment is shown in Fig.6.

The software is quite simple in terms of the development, with no extra settings for serial port. OMD800 device constantly analyzes the status of the power supplies sources connected to it. This information is stored in the internal device registers, from which it can be read by means of Modbus protocol, i.e., key parameters can be displayed, for example, on the computer [16].

In Modbus communication protocol, different codes are used for the requests to be sent to the involved device, but Modbus request of SwitchEnergy program is always the same, since the values of the same registers are read from the same involved device in each request.

Modbus survey is carried out by the program in the form of a message. The content of the received message depends on the current status of electrical switch connected to OMD device at that moment. For example, there is the response message to the request in case when the primary electric network has been disconnected due to power outage and the generator has been successfully started. SwitchEnergy program generates the textual explanation according to the register and the value in it, so it is not necessary for the student to know what does the value of each register mean.

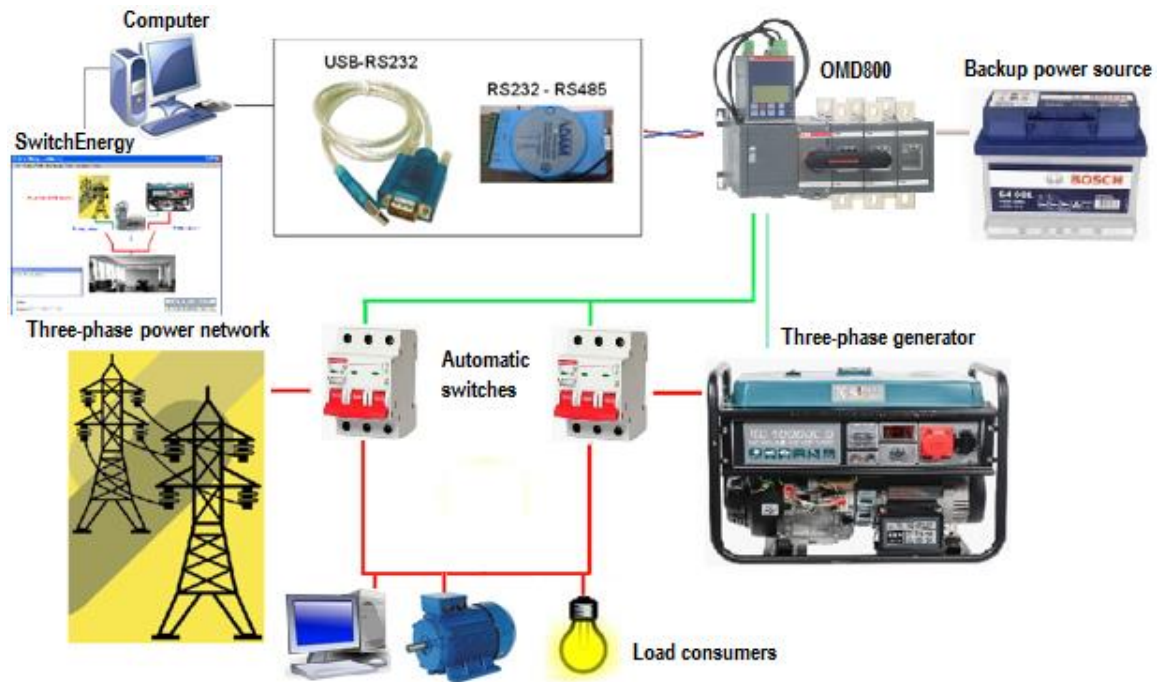


Figure6: Structural diagram of SwitchEnergy program's connection to the equipment.

When the student has selected the required status of OMD device as the operating mode of the program, the multi-stage process of connection creation starts. First, SwitchEnergy program surveys the operating system for available COM communication ports, then the ports are checked one by one by sending single Modbus request to this port. If there is the response to the request, the operating port is found, Modbus subprogram requests new values from OMD device every 500 milliseconds.

If OMD device receives the same values from the registers as in the previous reading, they are not re-entered into the program and the content of the ModbusData window is not changed. This prevents unnecessary program loading and flickering of the ModbusData window due to the high frequency of image generation. Modbus data window includes the number of sent and received messages, i.e. whether there is communication between the devices, even if the screen is not updated. Communication connection between SwitchEnergy program and OMD device is automatic and does not require additional actions from the student to update new values in the program [17].

4. Implementation of software functions

The general view of the proposed software is clear and visual, it contains three images with the diagram. Voltages on the electrical conductors, the statuses of the main switch, as well as the commands from OMD device to the generator and electrical networks. The view of the program, where the diagram with colored lines clarifies the information, is shown in Fig. 7.

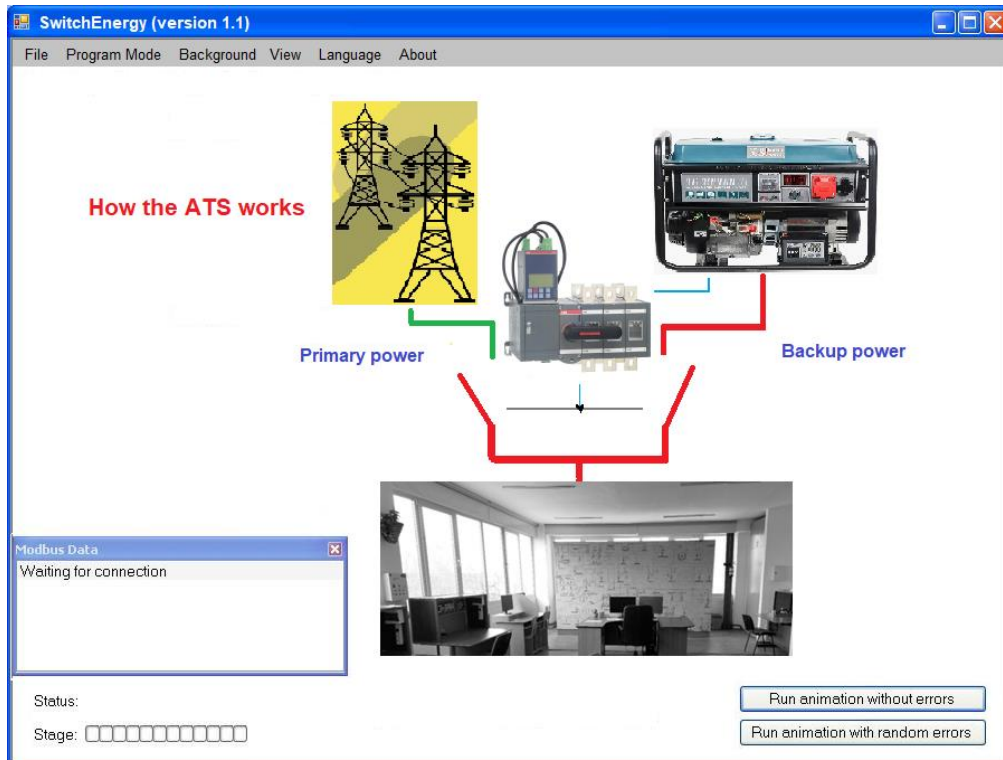


Figure 7: SwitchEnergy program view.

The images describe the conditions of primary and secondary links of the electrical network and the equipment connected to the load. Depending on the program mode, other components can be set in the main window. In accordance with the menu, it is possible to change the program mode, information presentation language, background images and visible elements.

4.1. Implementation of the function presentation mode

At the initial startup, the program automatically switches to the function presentation mode, making it possible to use the program independently of OMD device. In this mode, the program displays the status and operation of real OMD device in various situations. The program graphic consists of electrical circuits (red and green lines in the diagram), OMD device, and load switch that performs the switching. OMD device controls the generator of the second power supply source (blue line). In addition, the program has three images showing the primary power supply (top left), the secondary power supply (top right), and the training lab load connected to the power supply sources (bottom middle). Colors and brightness of images change depending on the activity of the device or image object. The view of the program window in the function presentation mode is shown in Fig. 8.

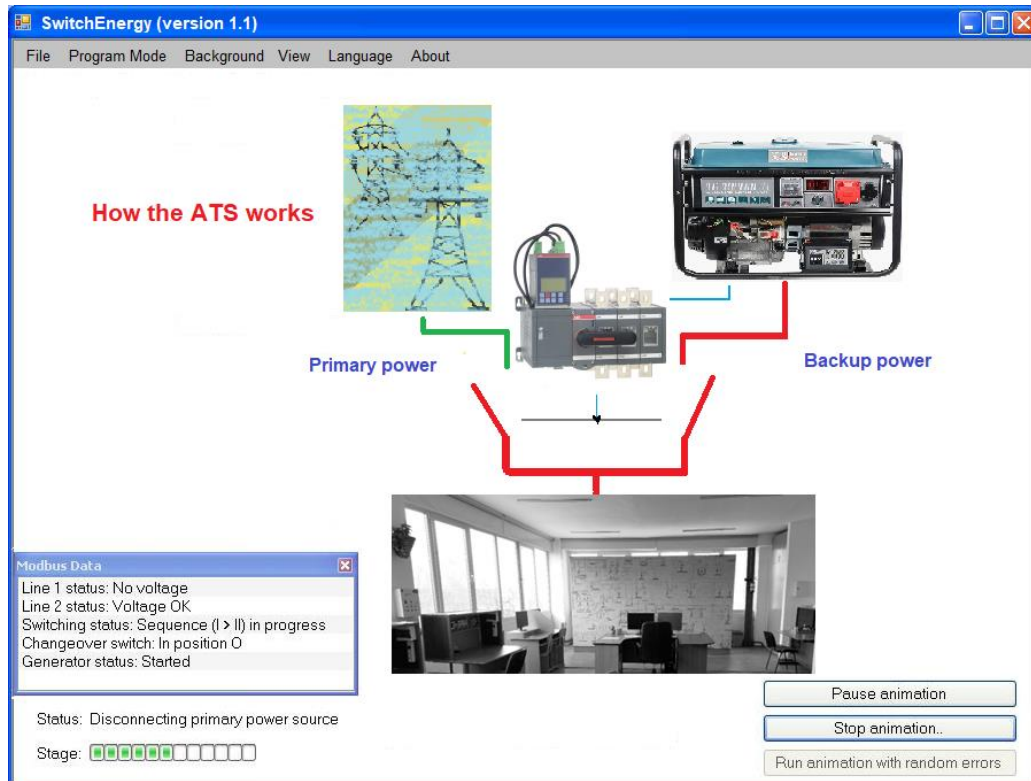


Figure 8: SwitchEnergy program window in function presentation mode.

In the function presentation mode it is possible to start animation that informs the student about the operation of OMD device in terms of electrical switch control and electrical network change. Graphical elements of the program in animation make it possible to display the following information on the monitor screen: whether power is supplied to the load, from which source power is supplied, in what status is the generator while controlling OMD device. The animation can be paused or completely stopped using the icons in the lower right part of the program. You can also choose the animation demonstration without error situations or with random error situations.

The possibility of OMD device application with different loads at arbitrary power source makes it possible for the student to switch all three images with the selection of the background presentation and the language of the program with the “mouse” marker in accordance with the task set in the laboratory work.

4.2. Implementation of the real device mode

The status of the real device is almost identical to the presentation device. When the student selects real device mode in the top menu, all icons in the lower right corner disappear from the user interface. Icons describing the animation process disappear from the lower left edge as well. At this point in the main program, Modbus data window is activated and another window “Status” providing information about the mode of the load switch [18, 19] is created, and the light indicator shows the receipt of electrical energy by the laboratory equipment (Fig. 9).

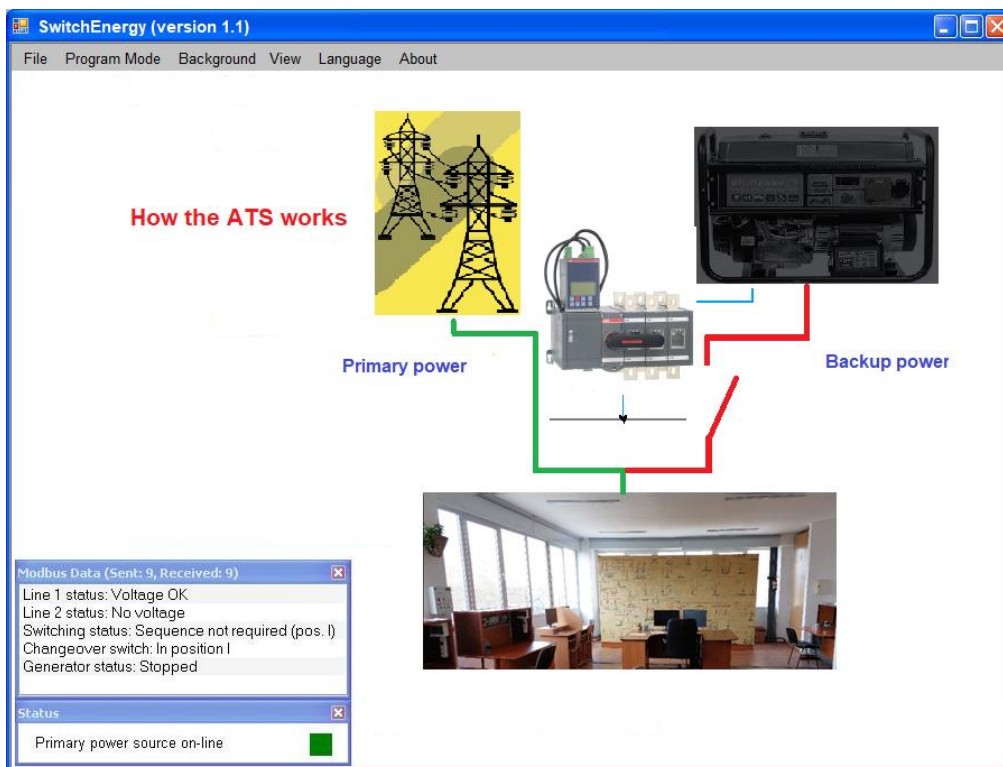


Figure 9: Switch Energy program window in real device mode

Режим представлення необхідного пристрою не вимагає від студента додаткових дій. Зв'язок з пристроєм OMD800 встановлюється автоматично, без налаштування додаткових опцій в програмі. Інформація в режимі реального часу про стан джерел живлення, подачі живлення на навантаження, положення електричного вимикача навантаження та стан генератора представлена в графічному вигляді програми, а також в режимі анімації.

The only difference from the animation mode in terms of diagram image is the display of changes of diagram lines at the moment when OMD device changes the parameters of power supply sources connected to it. The diagram is created without animation, sincw in this mode it is necessary to demonstrate the speed of the equipment's response to status changes with the least possible delay. The status change is recorded in the program with maximum delay of 0.5 seconds.

5. Conclusion

The oobjective of this project is to develop automated reserve input system based on the laboratory stand and software for communication with OMD800 switch controller produced by ABB Company for the implementation in the educational process at the Department of Electrical Engineering of Ternopil Ivan Puluj National Technical University j[20]. The developed software is very simple and supports illustrative and educational

application. The developed program has two modes of operation: with and without OMD device. The program's graphics makes it possible for the student to get acquainted with the functions and capabilities of OMD devices and additional functions, in particular, exchange animations and wallpapers, language selection is available (English, Ukrainian).

The connection of external equipment to OMD800 is designed in accordance with the requirements of SwitchEnergy program using Modbus protocol. Information transmitted by protocol about the current status of OMD device can be presented in the program both graphically and verbally. The animation mode has also been developed for the program. It makes it possible to present the functions of OMD device without physical connection to the device..

In the future, we are planning to continue our work on the project, in particular, to develop additional functions for the automatic connection of external electrical equipment supporting Modbus exchange protocol and to develop the user manual.

Acknowledgements

The authors of this paper express their gratitude to “LEP” LTD Company, in particular to Deputy Director Serhii Pankiv, engineers Vasyl Ivantsiv and Ivan Kapichovskyi, for their assistance in the laboratory stand development.

Special thanks to the Ternopil Design Bureau of Radio Engineering “Strila” for their important contribution to the development of the dispatch control simulator, particularly to Director O. Rafaliuk, Chief Engineer S. Piskun, and programmer S. Dudin.

The authors also express their gratitude to the students of the Faculty of Applied Information Technologies and Electrical Engineering who assisted in practical implementation of the project.

References

- [1] Ya. Koval, O. Bondarenko, M. Bolotnyi. Zabezpechennia nadiinosti elektropostachannia v suchasnykh lokalnykh elektrychnykh merezhakh, Vidnovliuvana enerhetyka ta enerhoefektyvnist u XXI stolitti, materialy XXIII mizhnarodnoi naukovo-praktychnoi konferentsii, Kyiv, 19-20 may 2022 pp. 69-71.[in Ukrainian].
- [2] F. Shcrabets, P. Krasovskyi, V. Berdnyk, Scientific Bulletin of National Mining University, 2017, Vol 2. – pp. 81-86.
- [3] Automatic control units, OMD800. Installation and operating instructions. ABB Oy, Protection and Connection, P.O. Box 622, FI-65101 Vaasa, Finland. <http://new.abb.com/low-voltage>.
- [4] Bogdan Orobchuk, Serhii Babiuk, Oleh Buniak, Ivan Sysak, Liubov Kostyk, Myroslav Nakonechniy, Yaroslav Filiuk, Development of an educational laboratory stand at the base fast-acting automatic reserve input, Scientific Journal of TNTU, 2023, Vol 112, pp. 12–25.

- [5] B.Orobchuk, O.Bunyak, Metodyka zastosuvannya systemy keruvannya rezhymamy elektropostachannya u navchalnomu protsesi. Materialy VII mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi "Pidvyshchennya rivnya efektyvnosti enerhospozhyvannya v elektrotekhnichnykh prystroyakh I systemakh" (Lutsk, 23.06.2018), M-vo osvity I nauky Ukrainy, Lutskyy natsion. tekhn. un-t, 2018, pp. 67-71. [in Ukrainian].
- [6] Avtomatyzovana systema dyspetcherskoho keruvannya "Strila". Tekhnichnyy opys I instruktsiya z ekspluatatsiyi. – Ternopil, 2010. [in Ukrainian].
- [7] B. Orobchuk, I. Sysak, S. Babiuk, T. Rajba, M. Karpinski, A. Klos-Witkowska, R. Szkarczyk, J. Gancarczy. Development of simulator automated dispatch control system for implementation in learning, 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). - IEEE, Buharest, vol.1, September 2017, pp. 210-214.
- [8] B.Orobchuk, S.Piskun, O.Rafalyuk. Vprovadzhennya system telemekhaniky keruvannya enerho-obyektamy v navchalnomu protsesi. III Vseukrayinska naukova-tekhnichna konferentsiya "Teoretychni ta prykladni aspekty radiotekhniky I prykladobuduvannya" (Ternopil, cherven 2017 r.), M-vo osvity I nauky Ukrainy, Tern. natsion. tekhn. un-t im. I. Pulyuya [ta in]. – Ternopil: TNTU, 2017. – pp. 224-226. [in Ukrainian].
- [9] B.Orobchuk, O.Buniak, S.Babiuk, I.Sysak, Design of an intelligent system to control educational laboratory equipment based on a hybrid mini-power plant. Eastern-European Journal of Enterprise Technologies, 2023, 2(9-122), pp. 59-72. (Scopus ISSN 1729-3774).
- [10] Modbus Technical Resources [Veb-sayt]. - Andover, Massachusetts, USA, 2023. - URL: <https://modbus.org/tech.php> (data zvernennya: 15.08.2023).
- [11] GitHub Copilot and Visual Studio 2022[Veb-sayt]. - URL:<https://visualstudio.microsoft.com/>
- [12] Modbus Poll User Guide[Veb-sayt]. - URL:<https://www.modbustools.com/mbpoll-user-manual.html>
- [13] MODBUS over serial line specification and implementation guide V1.02 [Veb-sayt]. - URL:<https://www.modbus.org/>
- [14] Automatic transfer switches, OTM_C_D_, Installation and operating instructions, ABB Oy, 1SCC303003M0202, rev. B
- [15] Ihor Bodnarchuk, Volodymyr Lisovyi, Olexandr Kharchenko, Iryna Galai: Adaptive Method for Assessment and Selection of Software Architecture in Flexible Techniques of Design. CSIT (1)2018:292-297
- [16] Diponkar Paul, Md. Shariful Islam, Syed Khalid Rahman. Performance evaluation and operation of Auto Load transfer switch, , International Journal of Renewable and Sustainable Energy 2013; 2(6): P. 205–211
- [17] B.Orobchuk, Laboratornyy kompleks dlya pobudovy system telekeruvannya I dyspetcherskoho upravlinnya v elektroenerhetytsi. Materialy XX naukovoyi konferentsiyi TNTU im. I. Pulyuya (Ternopil, traven 2017 r.), M-vo osvity I nauky Ukrainy, Tern. natsion. tekhn. un-t im. I. Pulyuya [ta in]. – Ternopil: TNTU, 2017. – pp. 170-171. [in Ukrainian].

- [18] Vymykachi navantazhennya, Ofitsiynyy predstavnyk ABB v Ukrayini [Veb-sayt]. - Kharkiv, 2023. - URL: <https://abbua.com.ua/nizkovoltne-obladnannya/vimikachi-navantazhennya-abb>
- [19] Reversyvni vymykachi navantazhennya. Ofitsiynyy predstavnyk ABB v Ukrayini [Veb-sayt]. - Kharkiv, 2023. - URL: <https://abbua.com.ua/nizkovoltne-obladnannya/vimikachi-navantazhennya-abb/reversivni-vimikachi-navantazhennya>.
- [20] B.Orobchuk, Dilovi ihry v navchal'nomu protsesi v yakosti instrumenta profesiynoyi pidhotovky mahistriv. VI Mizhnarodna naukovo-tekhnichna konferentsiya "Svitlotekhnika y elektroenerhetyka: istoriya, problemy, perspektyvy" (Ternopil - Yaremche, 30 sichnya – 2 lyutoho 2018 r.), M-vo osvity I nauky Ukrayiny, Tern. natsion. tekhn. un-t im. I. Pulyuya [ta in]. - Ternopil: TNTU, 2018. – pp. 95-96. [in Ukrainian].