

The Design and Operation of IEC 61850 – Based Transformer Measuring Device

Authors

Paweł Kopański
Désiré D. Rasolomampionona

Keywords

monitoring, transformer, IEC 61850

Abstract

The submitted paper presents the results of research on the development of a monitoring system for HV power transformers using the IEC 61850 communication standard. The characteristic feature of this system is its implementation on Linux-based programmable PLC. Data registration functionality, aggregation, archiving and data visualization are merged on one controller. Data registration is performed through built-in digital and analogue input cards, Ethernet and serial ports used for communication with external devices. Data aggregation and archiving is based on the MySQL database system, through data visualization and presentation thanks to an in-built HTML server with an implemented HTML web page that shows the most important parameters in real time. In addition, the implemented IEC 61850 stack enables a direct communication with the SCADA system. The stack has been implemented within the controller in such a way that the need for additional equipment, such as standard machine translation, is reduced. Both the device and all the algorithms developed in the Institute of Electrical Engineering allow collecting and recording the signals in real time, and moreover make a preliminary diagnosis of the transformer.

DOI: 10.12736/issn.2300-3022.2014106

1. Introduction

The constantly growing need for remote control of devices installed in the NPS has led to the idea of the development of a test and measurement device dedicated to the highest power transformers. The device is based on a Linux-enabled PLC, and meets the latest communications standards, including the IEC 61850 standard.

2. Design of the device

2.1. Hardware

The measuring device design is based on a Linux-enabled PLC with analogue (AI) and digital (DI) input cards. The analogue input cards collect data in the range of 4-20 mA from sensors installed on the transformer, such as: oil, core, and ambient temperatures. The digital inputs register trips of protection devices with NO/NC contacts, such as anti-explosion Buchholz relay and overcurrent protection of cooling fan motors. Additional measuring equipment, such as sensors of flammable gas content in the oil, or cooling system controllers are connected through an RS485 serial communication port. Measurements are taken and data read out most often in Modbus RTU/TCP standard. All these devices are installed in a measurement system cabinet on the transformer. A fibre link between the cabinet and the substation control and

monitoring system allows for remote viewing of the transformer parameters. An overall connection diagram of the control and measurement system is shown in Fig. 1. The use of two parallel PLCs enhances the system's reliability.

2.2. Software

With a view to the accomplishment of the measurement system's main objectives many Linux-based algorithms were developed to enable the collection of data from the transformer. Each algorithm implements a separate functionality. The element that binds all the algorithms and allows for their mutual communication is a MySQL database. Some programs, responsible for collecting data from devices, write data to the database, whereas other programs, responsible for visual rendering and transfer of data to supervisory systems, read data from the database. The interrelations between the programs are shown in Fig. 3.

The MySQL database is the fundamental operating element of the entire data recording, archiving and visual rendering algorithm. All records are written to, and read from, it. The database is organised in a number of tables, as shown in Fig. 2. The AI Signals and DI Signals tables store all records for data visual rendering or presentation. Data read from the controller's analogue and digital inputs is written directly to these tables. The Modbus

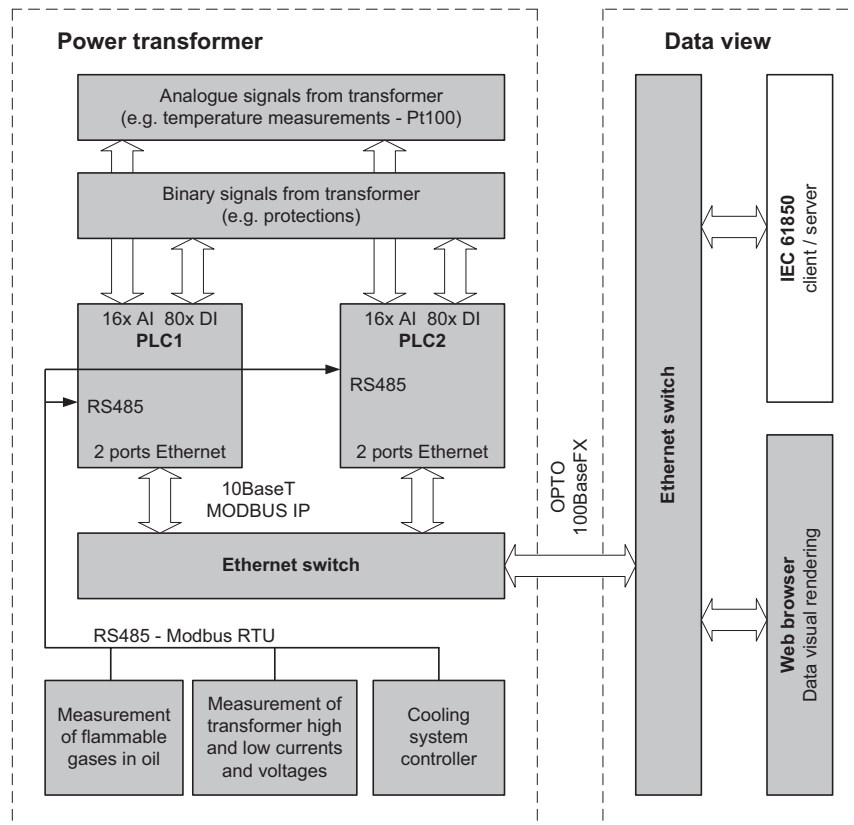


Fig. 1. Connection diagram of the transformer measuring system

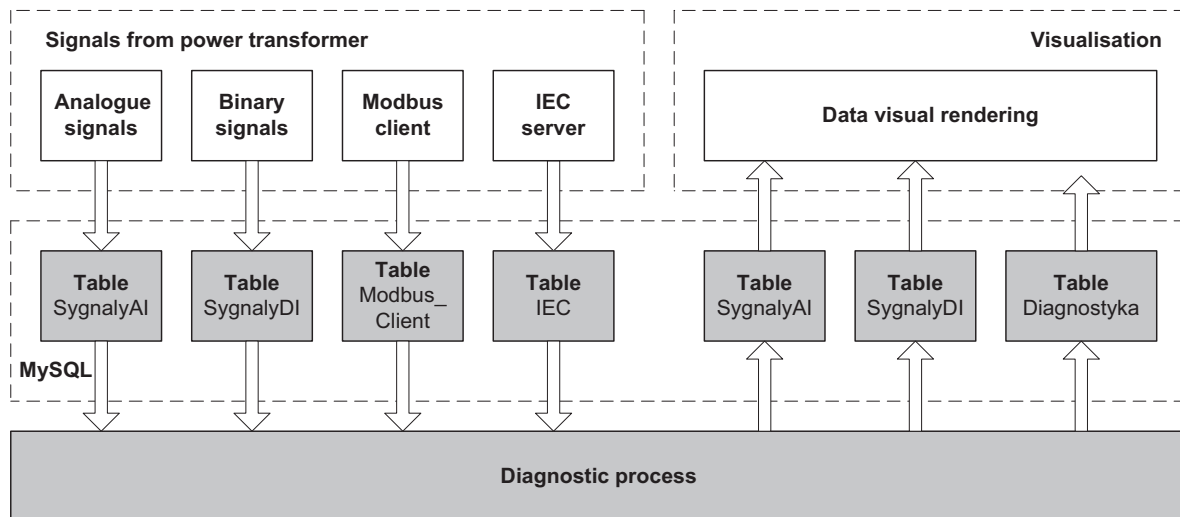


Fig. 2. Table structure in the MySQL database

RTU/TCP client that collects data from additional devices, such as water and hydrogen content in oil, transformers high and low currents, and cooling system control, operates in a separate thread. Data collected from these devices are first recorded in the device-specific tables, and then scaled in the diagnostic program and transferred to the output tables: AI Signals or DI Signals. The diagnostic thread tracks all details read from, and written to, the database, and on this basis it generates messages on the transformer's technical condition and limit value exceedances.

A timer set to 1 s cyclically reads the records last saved in databases containing instantaneous values read from devices. After all readings the data is compared with signals recorded in the previous cycle. If any analogue value in the input tables has changed by more than the allowable threshold, it is adjusted, rescaled, and written to readout tables in the database. Next, binary signals are checked. If any signal has changed its status, a warning or notice message is generated. If, for example, it is the binary signal from a safety relief valve, it will generate a "safety

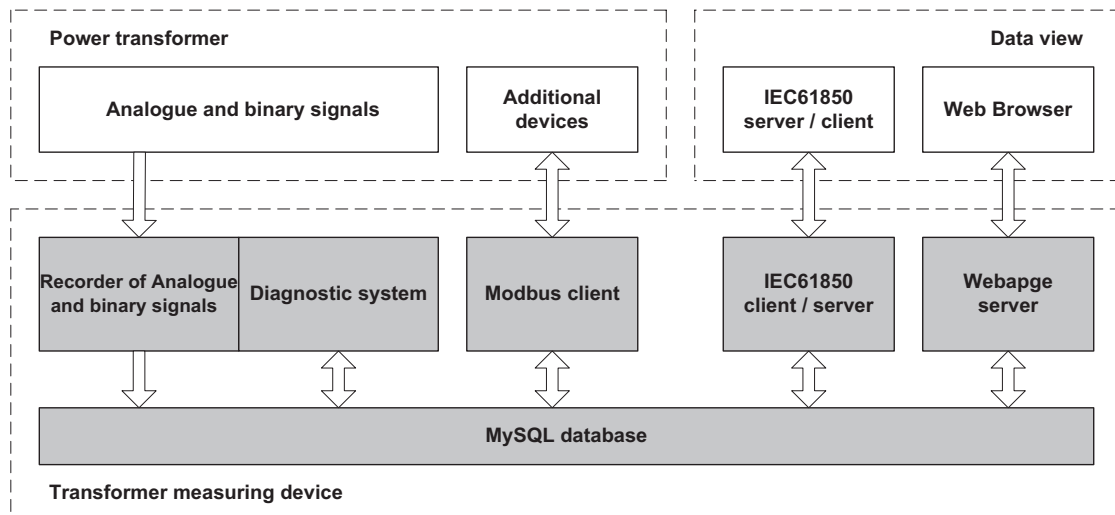


Fig. 3. Interrelations between the controller software

```

53     readVariable    Relay/MMXU$CF$ClockTOD
54 }
55
56 define testReadVar1 {
57     #   init
58     readVariableSql KoncPOD/MMXU0$MX$PhV1$phsA$cVal$mag$f f0
59     readVariableSql KoncPOD/MMXU0$MX$PhV1$phsB$cVal$mag$f f1
60     readVariableSql KoncPOD/MMXU0$MX$PhV1$phsC$cVal$mag$f f2
61     readVariableSql KoncPOD/MMXU0$MX$PhV2$phsA$cVal$mag$f f3
62     readVariableSql KoncPOD/MMXU0$MX$PhV2$phsB$cVal$mag$f f4
63     readVariableSql KoncPOD/MMXU0$MX$PhV2$phsC$cVal$mag$f f5
64     readVariableSql KoncPOD/MMXU0$MX$TotW1$mag$f f6
65     readVariableSql KoncPOD/MMXU0$MX$TotVAr1$mag$f f7
66     #   readVariableSql Relay/MMXU0$MX$VA$PhsAf f8
67     #   readVariableSql Relay/PBRO$ST$Out b0
68     #   readVariableSql Relay/PDOC$ST$Out b1
69     #   conclude
70 }
71

```

Fig. 4. Example configuration of the IEC 61850 client

relief valve tripping” alarm message. The message will be saved to the database of transformer diagnostics. The other algorithms shown in Fig. 2 enter data into the input tables. One of the programs implementing such an algorithm is the Modbus client. It enables Modbus TCP or Modbus RTU communication with all additional devices installed on the transformer. For the purpose of this program the entire Modbus client library was developed. The library contains the most popular functions used for reading data, such as: Read Coils, Read Single Coils, Read Input Registers, and Read Holding Registers. So read data from the devices is stored in the database’s device-specific tables. The last program for entering data into the database is the program for IEC 61850

communication with the substation’s supervisory station. It allows two-way communication so that data can be transmitted from the monitoring system to the supervisory system, and vice versa. The activated IEC client allows initiating communication with the supervisory system, to download data and save it to the database. Fig. 4 shows an example configuration and implementation of the IEC 61850 client installed in the controller. High and low transformer voltage variables from KoncPOD/MMXU0 MX \$ \$ \$ PhV1 namespace are saved in floating point format in the IEC table, under addresses 0, 1, 2, etc. In a similar way the binary signals can be mapped, by naming them b0, b1 etc.

```

D:\Lincon\comboiec_mysql\data.ini - Notepad++
Plik  Edycja  Szukaj  Widok  Format  Składnia  Ustawienia  Makra  Uruchom  Pluginy  Okno  ?
Polecenia MySQL.txt  lincon_all.c  mb_client.c  client.ini  KONC.sim  monitoring.css  monitoring.php

1  #prad gorny - faza A
2  ReadSql 0 = Write AT1/MMXUGN$MX$A$phsA$cVal$mag$i -int -1
3  ReadSql 0 = Write AT1/MMXUGN$MX$A$phsA$cVal$mag$f -float -100
4  #prad gorny - faza B
5  ReadSql 1 = Write AT1/MMXUGN$MX$A$phsB$cVal$mag$i -int -1
6  ReadSql 1 = Write AT1/MMXUGN$MX$A$phsB$cVal$mag$f -float -100
7  #prad gorny - faza C
8  ReadSql 2 = Write AT1/MMXUGN$MX$A$phsC$cVal$mag$i -int -1
9  ReadSql 2 = Write AT1/MMXUGN$MX$A$phsC$cVal$mag$f -float -100
10 #prad dolny - faza A
11 ReadSql 3 = Write AT1/MMXUDN$MX$A$phsA$cVal$mag$i -int -1
12 ReadSql 3 = Write AT1/MMXUDN$MX$A$phsA$cVal$mag$f -float -100
13 #prad dolny - faza B
14 ReadSql 4 = Write AT1/MMXUDN$MX$A$phsB$cVal$mag$i -int -1
15 ReadSql 4 = Write AT1/MMXUDN$MX$A$phsB$cVal$mag$f -float -100
16 #prad dolny - faza C
17 ReadSql 5 = Write AT1/MMXUDN$MX$A$phsC$cVal$mag$i -int - 1
18 ReadSql 5 = Write AT1/MMXUDN$MX$A$phsC$cVal$mag$f -float -100
19 #wejscie dwustanowe DIO - LinCon
20 ReadSql 0 = Write AT1/YPTR1$ST$OpUnV$stVal -bool
21 ReadSql 1 = Write AT1/YPTR1$ST$OpOvV$stVal -bool
22 #
    
```

Fig. 5. Example configuration of the IEC 61850 server

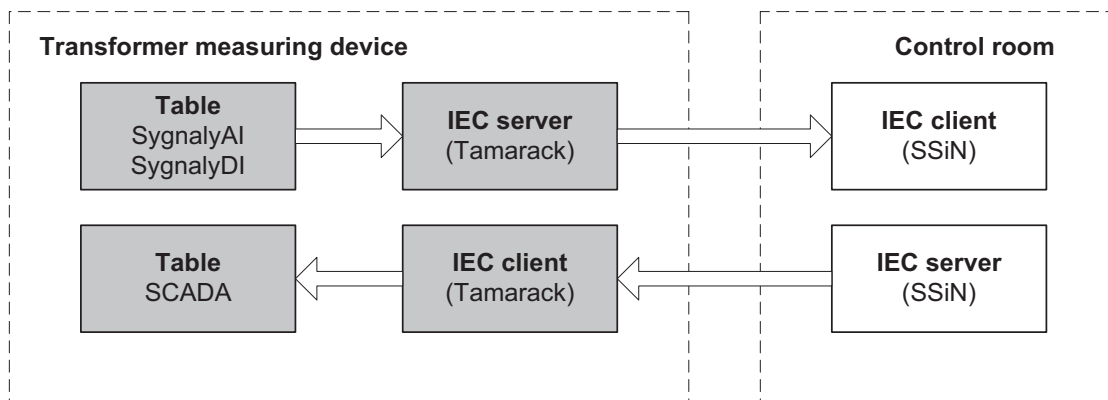


Fig. 6. Client – server interconnection structure in the IEC model

The IEC Server enables communication in the other direction, i.e. retrieves data from the database, and presents it in the IEC model structure. Fig. 5 shows an example configuration of the server. ReadSQL 0 signifies the instruction to download the zero variable from the AI Signals table. The Write AT1/MMXUGN\$MX\$A\$phsA\$cVal\$mag\$i command presents it in the IEC model namespace relating to currents measured on the transformer high side. The rest of the record is the data type (integer, float, or bool). This is how the entire table of signals made available to SCADA systems can be easily mapped. The interrelations of the IEC server and client installed in the controller are shown in Fig. 6.

The web server implemented in the controller runs a webpage allowing the visual rendering of data collected from all devices. The data is read from the database's output tables AI Signals and DI Signals. The entire database reading engine is written in PHP. A separate algorithm that utilises AJAX libraries enables asynchronous uploading of the webpage content. This mechanism allows refreshing only those fragments that contain read-out and changed values. The webpage's layout was designed in CSS. The home page screenshot is shown in Fig. 7. It contains the most important details of the transformer load, oil temperature, tap changer position, and the last generated messages.

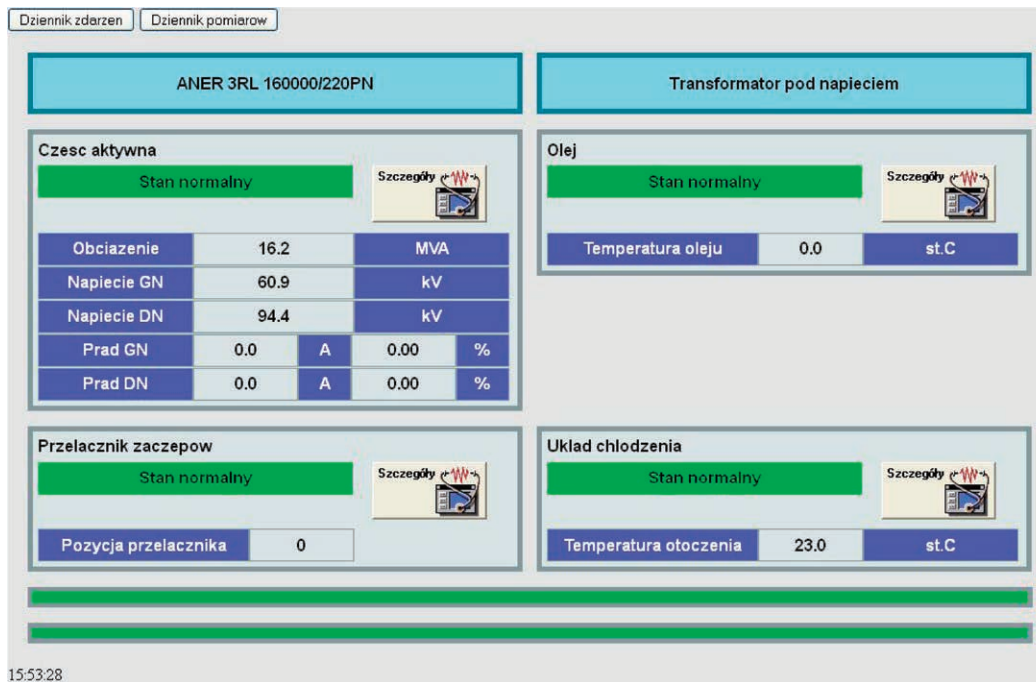


Fig. 7. The webpage for visual rendering of data collected from transformer

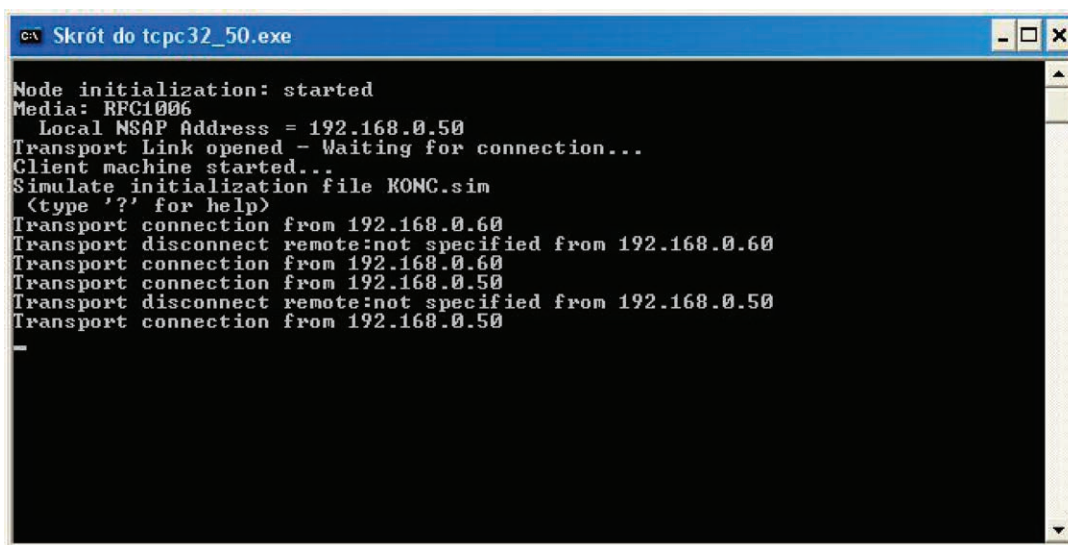


Fig. 8. Screenshot of the program simulating the IEC 61850 server in substation control and supervision system

3. Tests

3.1. Test bench

The algorithms were tested in a laboratory bench, with a transformer performance simulator and additional programs for simulation of measuring devices' performance. Analogue and digital signals fed to input cards in the tested device were generated by a PC. Actual values of such signals as core temperature or protection tripping were simulated through the software controlling signals at the output cards. Devices such as in-oil hydrogen and water sensor, cooling system controller, and current meters on transformer high and low sides were simulated by the Modbus

server software. The simulator, developed specifically for this purpose, allocates appropriate values in the address space, as in the case of real devices.

In the IEC 61850 server that simulated the substation control and supervision system a model was implemented that contained high and low transformer voltages.

The IEC client embedded in the controller retrieved the values from the model and saved to the database. Screenshot of the server performance simulating program is shown in Fig. 8. The program used to view data presented by the IEC server in the controller, which simulated the SCADA systems performance, was IEC Browser.

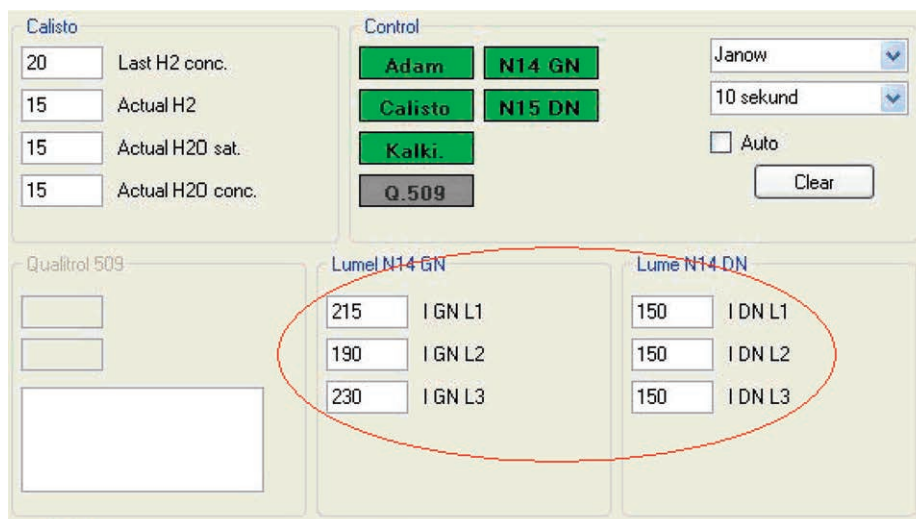


Fig. 9. Screenshot of the Modbus server simulation programme

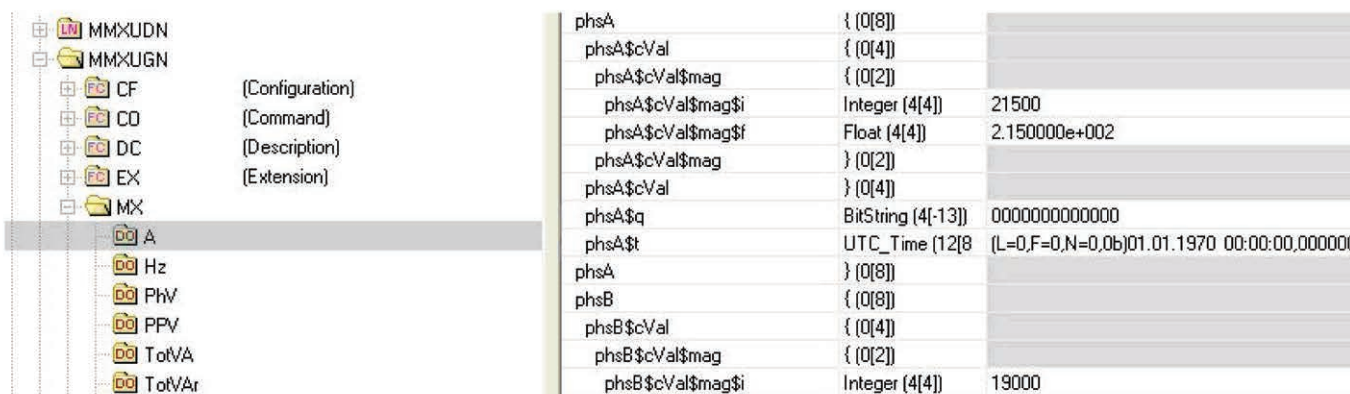


Fig. 10. Screenshot of a fragment of the IEC 61850 client simulation programme

3.2. Tests

To a large extent the system tests focused on verifying the data flow and IEC stacks compatibility. For example, one of the many tests was to check whether the data read from measuring devices are properly seen in the visual rendering systems.

As shown in Fig. 9, the values of 215 A, 190 A, and 230 A were entered to the program that simulated the current meter at the transformer HV side. The program inserted the values under specific addresses in the Modbus protocol, in accordance with the actual device documentation.

The controller’s task was to automatically establish communication with the Modbus server, to retrieve the values, rescale them, and save to the database.

The web server, retrieving data from the database at specific time intervals, was supposed to visually render them in the appropriate right panel. An excerpt from the webpage presenting the retrieved data is shown in Figure 11. In addition, if an allowable limit was exceeded, a warning would have been generated and displayed on the main visual rendering panel.

Prady		
Prad fazowy L1 GN	215.0	A
Prad fazowy L2 GN	190.0	A
Prad fazowy L3 GN	230.0	A

Moc		
Czynna	27.4	MW
Bierna	55.1	kVar
Pozorna	51.1	MVA

Fig. 11. Screenshot of a fragment of the webpage showing currents and power

The IEC server run in a separate thread retrieved data saved in the database and forward them for readout in SCADA systems. A screenshot of a fragment of the program presenting high side transformer current details retrieved by the IEC 61850 is shown in Fig. 10.

Besides IEC Browser, which simulated SCADA systems performance, Tamarack client and IEC Scout software was also used in the tests. All programs had successfully passed tests of compatibility with the IEC server implemented in the controller. As shown in the included illustrations (Fig. 9 – Fig. 11), values simulated in the Modbus server were correctly read in the webpage visualization, as well as in the program acting as the IEC 61850 client.

REFERENCES

1. Kowalik R., Kopański P., Glik K., "Autotransformer monitoring system" International Conference on Renewable Energies And Power Quality (ICREPQ '10), pp. 23–25 March 2010, Grenada, Spain, p. 692.
 2. Kowalik R., Januszewski M., Example results of electrical power system protection devices interoperability tests using IEC 61850 standard, *Electrical Review* 2008, No. 9, pp. 114–119.
 3. Kowalik R., Januszewski M., Coordination of protections through the exchange of binary signals between power system substations in the IEC 61850 protocol, *Electrical Review* 2009, No. 6, pp. 195–201.
-

Paweł Kopański

Warsaw University of Technology | Twelve Electric sp. z o.o.

e-mail: pawel.kopanski@ien.pw.edu.pl

Graduated from the Faculty of Electrical Engineering, Power System Automation Department, of Warsaw University of Technology (2009). For several years has participated in the development and implementation of IT systems. For the most part these are measuring systems for data concentration, aggregation, and archiving. Currently Head of IT Department at Twelve Electric sp. z o.o. [Ltd.].

Désiré D. Rasolomampionona

Warsaw University of Technology

e-mail: desire.rasolomampionona@ien.pw.edu.pl

Since 1994 a researcher/teacher in the Electrical Power Engineering Institute of Warsaw University of Technology. He is currently the head of the Department of Power Apparatus, Protection and Control. His research interests are focused mainly on the issues of electrical power automation, control of the power system operation and the use of telecommunications and modern information technologies in electrical power engineering.

This is a supporting translation of the original text published in this issue of "Acta Energetica" on pages 72–78. When referring to the article please refer to the original text.

PL

Konstrukcja i działanie urządzenia pomiarowego transformatorów wykorzystującego IEC 61850

Autorzy

Paweł Kopański
Désiré D. Rasolomampionona

Słowa kluczowe

monitoring, transformator, IEC 61850

Streszczenie

Artykuł zawiera wyniki badań nad opracowaniem systemu monitoringu transformatorów najwyższych mocy wykorzystującego standard IEC 61850. Cechą charakterystyczną opisywanego systemu jest jego implementacja na programowalnym sterowniku PLC, bazującym na systemie Linux. W jednym sterowniku została połączona funkcjonalność rejestracji, agregacji, archiwizacji i wizualizacji danych.

1. Wstęp

Stale rozwijająca się potrzeba prowadzenia zdalnej kontroli nad urządzeniami zainstalowanymi w KSE doprowadziła do koncepcji opracowania urządzenia kontrolno-pomiarowego, dedykowanego transformatorom najwyższych mocy. Urządzenie oparte jest na sterowniku PLC, pracującym na platformie Linux, i spełnia najnowsze standardy telekomunikacyjne, w tym normę IEC 61850.

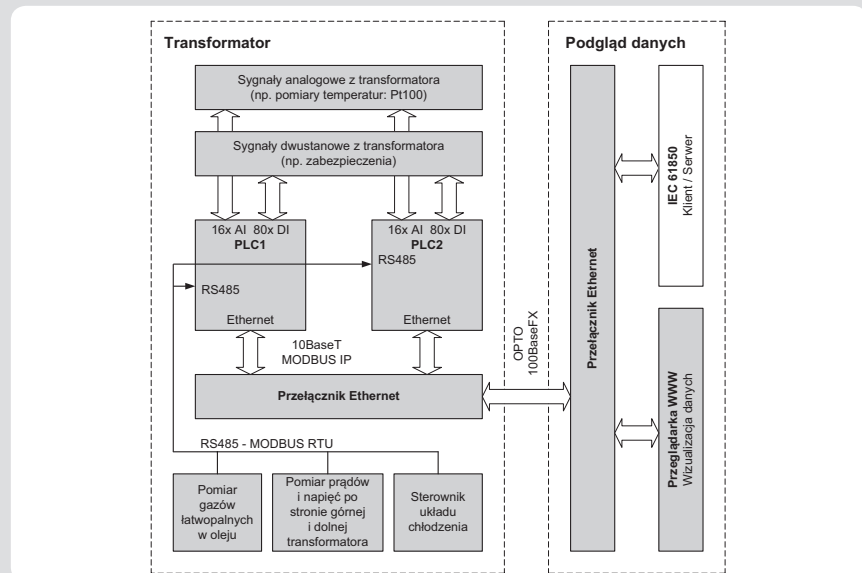
2. Budowa urządzenia

2.1. Część sprzętowa

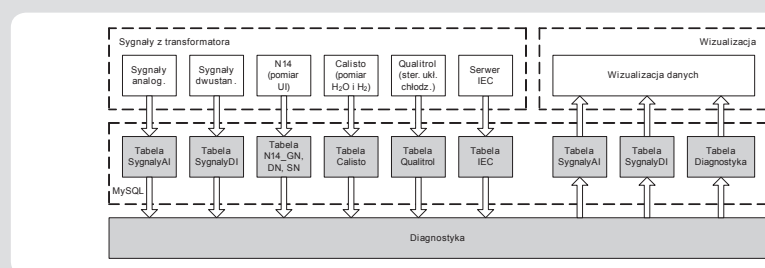
Budowa urządzenia pomiarowego została oparta na sterowniku PLC, wyposażonym w karty wejść analogowych (AI) oraz dwustanowych (DI), pracującym w środowisku Linux. Karty wejść analogowych umożliwiają zbieranie danych w standardzie 4–20 mA z czujników zlokalizowanych na transformatorze, takich jak: czujniki temperatury oleju, rdzenia czy otoczenia. Wejścia dwustanowe umożliwiają rejestrację zadziałania urządzeń zabezpieczeniowych, wyposażonych w styki zwierzno-rozwiernie, jak np. przekaźnik przeciwwybuchowy Buchholza czy zabezpieczenia nadprądowe silników wentylatorów chłodnic. Przez port transmisyjny szeregowy RS485 podłącza się dodatkowe urządzenia pomiarowe typu: czujnik zawartości gazów łatwopalnych w oleju czy sterownik układu chłodzenia. Pomiar i odczyt danych najczęściej odbywa się w standardzie Modbus RTU/TCP. Wszystkie te urządzenia są zainstalowane w szafie układu pomiarowego na transformatorze. Połączenie światłowodowe między szafą pomiarową a systemem sterowania i nadzoru stacji umożliwia zdalny podgląd parametrów transformatora. Ogólny schemat podłączenia systemu kontrolno-pomiarowego został przedstawiony na rys. 1. Zastosowanie dwóch sterowników PLC pracujących równolegle zwiększa niezawodność układu.

2.2. Oprogramowanie

Na potrzeby realizacji głównych założeń układu pomiarowego zostało zaprojektowanych wiele algorytmów pracujących w środowisku Linux, umożliwiających zbieranie danych z transformatora. Każdy algorytm służy do realizacji oddzielnych



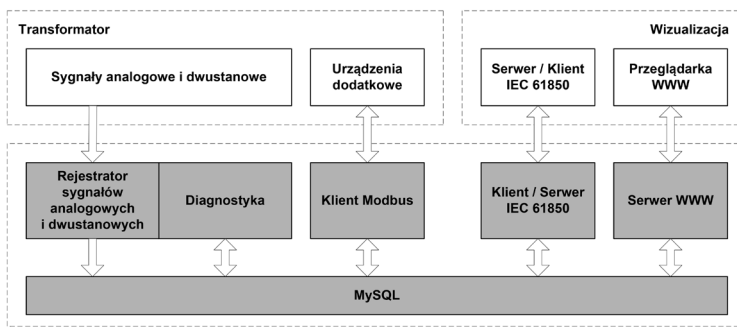
Rys. 1. Schemat podłączenia systemu pomiarowego transformatora



Rys. 2. Struktura tabel w bazie MySQL

funkcji. Elementem spajającym wszystkie algorytmy, pozwalającym na wzajemną komunikację, jest baza danych MySQL. Część programów odpowiedzialna za pobieranie danych z urządzeń zapisuje dane do bazy. Z kolei pozostałe programy odpowiedzialne za wizualizację i przesył danych do systemów nadzorczych odczytują dane z bazy. Wzajemne powiązanie opracowanych programów zostało pokazane na rys. 3.

Baza danych MySQL jest podstawowym elementem pracy całego algorytmu rejestracji, archiwizacji i wizualizacji danych. Do niej są zapisywane i odczytywane wszystkie rekordy. Struktura bazy została podzielona na wiele tabel, zgodnie z rys. 2. W tabelach oznaczonych jako SygnalyAI oraz SygnalyDI przechowywane są wszystkie rekordy przeznaczone do wizualizacji lub prezentacji danych. Dane odczytywane z wejść analogowych oraz dwustanowych sterownika są



Rys. 3. Wzajemne powiązanie uruchomionych na sterowniku programów

```

53 readVariable Relay/MMXU0$CF$ClockTOD
54 }
55
56 define testReadVar1 {
57 # init
58 readVariableSql KoncPOD/MMXU0$MX$PhV1$phsA$cVal$mag$f #0
59 readVariableSql KoncPOD/MMXU0$MX$PhV1$phsB$cVal$mag$f #1
60 readVariableSql KoncPOD/MMXU0$MX$PhV1$phsC$cVal$mag$f #2
61 readVariableSql KoncPOD/MMXU0$MX$PhV2$phsA$cVal$mag$f #3
62 readVariableSql KoncPOD/MMXU0$MX$PhV2$phsB$cVal$mag$f #4
63 readVariableSql KoncPOD/MMXU0$MX$PhV2$phsC$cVal$mag$f #5
64 readVariableSql KoncPOD/MMXU0$MX$TotW1$mag$f #6
65 readVariableSql KoncPOD/MMXU0$MX$TotV1$mag$f #7
66 # readVariableSql Relay/MMXU0$MX$VA$PhsAf #8
67 # readVariableSql Relay/FBRO$ST$Out b0
68 # readVariableSql Relay/PDOC$ST$Out b1
69 # conclude
70 }
71
    
```

Rys. 4. Przykład konfiguracji klienta IEC 61850

```

1 #prad gorny - faza A
2 ReadSql 0 = Write AT1/MMXUGN$MX$A$phsA$cVal$mag$f -int -1
3 ReadSql 0 = Write AT1/MMXUGN$MX$A$phsB$cVal$mag$f -float -100
4 #prad gorny - faza B
5 ReadSql 1 = Write AT1/MMXUGN$MX$A$phsB$cVal$mag$f -int -1
6 ReadSql 1 = Write AT1/MMXUGN$MX$A$phsC$cVal$mag$f -float -100
7 #prad gorny - faza C
8 ReadSql 2 = Write AT1/MMXUGN$MX$A$phsC$cVal$mag$f -int -1
9 ReadSql 2 = Write AT1/MMXUGN$MX$A$phsC$cVal$mag$f -float -100
10 #prad dolny - faza A
11 ReadSql 3 = Write AT1/MMXUDN$MX$A$phsA$cVal$mag$f -int -1
12 ReadSql 3 = Write AT1/MMXUDN$MX$A$phsB$cVal$mag$f -float -100
13 #prad dolny - faza B
14 ReadSql 4 = Write AT1/MMXUDN$MX$A$phsB$cVal$mag$f -int -1
15 ReadSql 4 = Write AT1/MMXUDN$MX$A$phsC$cVal$mag$f -float -100
16 #prad dolny - faza C
17 ReadSql 5 = Write AT1/MMXUDN$MX$A$phsC$cVal$mag$f -int -1
18 ReadSql 5 = Write AT1/MMXUDN$MX$A$phsC$cVal$mag$f -float -100
19 #wejście dwustanowe DIO - LinCon
20 ReadSql 0 = Write AT1/YPTR1$ST$OpUnV$stVal -bool
21 ReadSql 1 = Write AT1/YPTR1$ST$OpOvV$stVal -bool
22 #
    
```

Rys. 5. Przykład konfiguracji serwera IEC 61850

w tabelach odpowiednich dla danego urządzenia, a następnie w programie służącym do diagnostyki są skalowane i przenieszone do tabel wyjściowych: SygnałyAI lub SygnałyDI.

Wątek do diagnostyki śledzi wszystkie odczytywane i zapisywane w bazie danych informacje i na tej podstawie generuje komunikaty o stanie technicznym transformatora oraz przekroczeniach dopuszczalnych wartości. Timer ustawiony na 1 s cyklicznie odczytuje ostatnie zapisane rekordy w bazach zawierających wartości chwilowe odczytane z urządzeń. Po dokonaniu wszystkich odczytów dane są porównywane z sygnałami zapisanymi w poprzednim cyklu. Jeśli w tabelach wejściowych nastąpiła zmiana którejkolwiek z wartości analogowych o wartość przekraczającą dopuszczalny próg, następuje dopasowanie, przeskalowanie i zapis do tabel odczytu w bazie danych. Następnie sprawdzane są sygnały dwustanowe. W chwili, gdy któryś z sygnałów zmieni swój stan, generowany jest komunikat ostrzegawczy lub informacyjny. Jeśli przykładowo będzie to sygnał dwustanowy, pochodzący od zaworu upustowego bezpieczeństwa, zostanie wygenerowany komunikat alarmowy o „Zadziałaniu zaworu upustowego bezpieczeństwa”. Komunikat zostanie zapisany do bazy danych z diagnostyką transformatora.

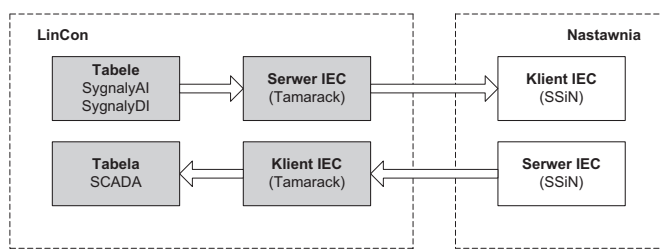
Pozostałe algorytmy przedstawione na rys. 2 służą do wprowadzania danych do tabel wejściowych. Jednym z takich algorytmów jest klient Modbus. Program umożliwia komunikację z wszystkimi urządzeniami dodatkowymi, umieszczonymi na transformatorze w standardzie Modbus TCP lub Modbus RTU. Na potrzeby tego programu została opracowana cała biblioteka klienta Modbus. Biblioteka zawiera najpopularniejsze funkcje wykorzystywane przy odczycie danych, takie jak: Read Coils, Read Single Coils, Read Input Registers oraz Read Holding Registers. Tak odczytane z urządzeń informacje są zapisywane w bazie danych do tabel odpowiednich dla każdego urządzenia.

Ostatnim programem służącym do wprowadzania danych do bazy danych jest program do komunikacji z systemem nadzorczym stacji w standardzie IEC 61850. Program umożliwia obustronną komunikację w taki sposób, że dane mogą być przesyłane zarówno od systemu monitoringu do systemu nadzorczego, jak i na odwrót. Uruchomiony klient IEC umożliwia zainicjowanie komunikacji z systemem nadzorczym, pobranie danych oraz ich zapis do bazy danych. Na rys. 4 znajduje się przykład konfiguracji oraz implementacji klienta modelu IEC 61850, zainstalowanego w sterowniku. Zmienne z przestrzeni nazw KoncPOD/MMXU0\$MX\$PhV1\$, odnoszące się do napięć w stronie górnej oraz dolnej transformatora, są zapisywane w formacie zmiennoprzecinkowym (ang. float) w tabeli IEC, pod adresami 0, 1, 2 itd. W podobny sposób można zamapować sygnały dwustanowe, przyporządkowując im nazwy: b0, b1 itd.

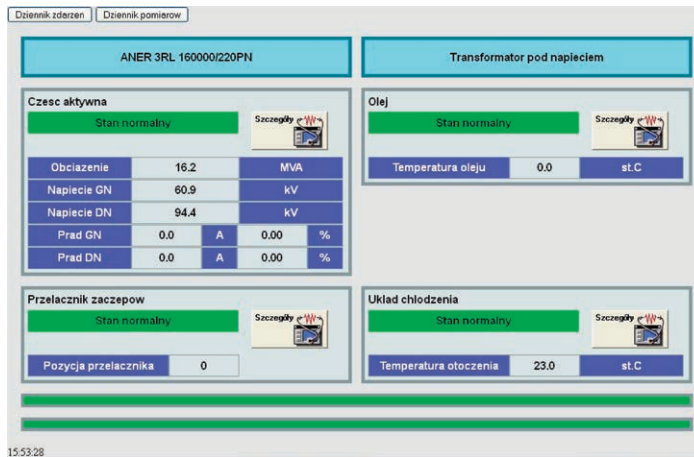
Serwer IEC umożliwia komunikację w drugą stronę, czyli pobranie danych z bazy oraz wystawienie ich w strukturze modelu IEC.

bezpośrednio zapisywane w tych tabelach. Klient Modbus RTU/TCP, służący do pobierania danych z urządzeń dodatkowych, jak np. czujnik wody i wodoru w oleju, miernik

prądu po stronie górnej i dolnej transformatora lub sterownik układu chłodzenia, pracuje w oddzielnym wątku. Dane pobierane z tych urządzeń są najpierw zapisywane



Rys. 6. Struktura wzajemnego powiązania, klient – serwer w modelu IEC



Rys. 7. Wygląd strony WWW do wizualizacji danych zebranych z transformatora

```

Skrót do tcp32_50.exe
Node initialization: started
Media: RFC1006
Local NSAP Address = 192.168.0.50
Transport Link opened - Waiting for connection...
Client machine started...
Simulate initialization file KOMC.sim
<type '?' for help>
Transport connection from 192.168.0.60
Transport disconnect remote: not specified from 192.168.0.60
Transport connection from 192.168.0.60
Transport disconnect remote: not specified from 192.168.0.50
Transport connection from 192.168.0.50

```

Rys. 8. Wygląd programu symulującego pracę serwera IEC 61850 w SSiN

Na rys. 5 znajduje się przykład konfiguracji serwera. ReadSQL 0 oznacza instrukcję dla pobrania zmiennej zerowej z tabeli SygnalyAI. Natomiast komenda Write AT1/MMXUGN\$MX\$A\$ wystawia ją w modelu IEC w przestrzeni nazw odnoszącej się do pomiarów prądów po stronie górnej transformatora. Dalsza część zapisu oznacza typ danych (integer – wartość całkowita, float – wartość zmiennoprzecinkowa, bool – typ zmiennej dwustanowej). W ten sposób można łatwo zmapować całą tablicę sygnałów udostępnianych systemom SCADA.

Wzajemne powiązanie serwera oraz klienta IEC, zainstalowanych w sterowniku, zostało pokazane na rys. 6.

Uruchomiony na sterowniku serwer WWW umożliwia zaimplementowanie strony do wizualizacji danych, zebranych z wszystkich urządzeń. Dane są czytane z bazy danych, z tabel wyjściowych SygnalyAI oraz SygnalyDI. Cały silnik do odczytu bazy danych został napisany w języku PHP. Oddzielny algorytm wykorzystujący biblioteki AJAX umożliwia asynchroniczne ładowanie zawartości strony. Taki mechanizm pozwala na odświeżenie tylko tych fragmentów, które zawierają odczytane i zmieniane wartości. Strona wizualna została zaprojektowana w CSS. Wygląd strony startowej, został przedstawiony na rys. 7. Znajdują się tu najważniejsze informacje o obciążeniu transformatora, temperaturze

oleju, pozycji przełącznika zaczerwów oraz ostatnie wygenerowane komunikaty.

3. Testy

3.1. Stanowisko do testowania

Testy opracowanych algorytmów były przeprowadzane na stanowisku laboratoryjnym przy wykorzystaniu symulatora pracy transformatora oraz programów dodatkowych, pełniących rolę symulatorów pracy urządzeń pomiarowych.

Sygnały analogowe oraz dwustanowe, doprowadzone na karty wejściowe w testowanym urządzeniu, wymuszane były przez komputer klasy PC. Poprzez oprogramowanie sterujące sygnałami na wyjściach kart symulowane były rzeczywiste wartości, takie jak np. temperatura rdzenia czy zadziałanie zabezpieczeń. Pracę takich urządzeń, jak czujnik wody i wodoru w oleju, sterownik układem chłodzenia czy mierniki prądów po stronie górnej i dolnej transformatora, symulowane były poprzez oprogramowanie będące serwerem Modbus. Napisany specjalnie do tego celu symulator wystawiał odpowiednie wartości w przestrzeni adresowej, takiej samej jak w przypadku rzeczywistych urządzeń.

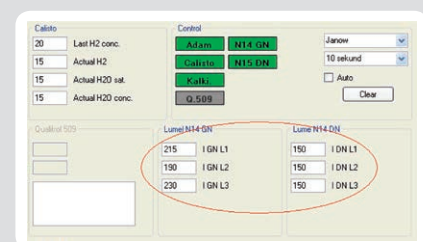
W serwerze IEC 61850, symulującym pracę Systemu Sterowania i Nadzoru Stacji, zaimplementowany został model zawierający m.in. wartości napięć strony górnej i dolnej transformatora. Poprzez zaszyty w sterowniku klient IEC wartości te były pobierane z modelu i zapisywane w bazie danych. Wygląd programu symulującego pracę serwera został przedstawiony na rys. 8. Programem używanym do podglądu danych wystawianych przez serwer IEC w sterowniku, symulującym pracę systemów SCADA, był IEC Browser.

3.2. Przebieg testów

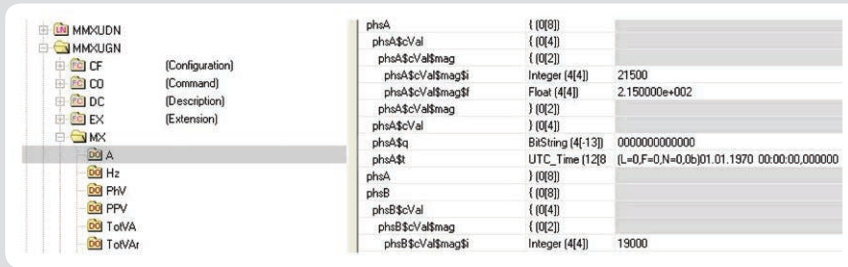
W głównej mierze testy systemu zostały skupione na sprawdzeniu poprawności przepływu danych i kompatybilności stosów IEC.

Dla przykładu jednym z wielu testów, jaki został przeprowadzony, było sprawdzenie, czy dane odczytywane z urządzeń pomiarowych będą prawidłowo widziane w układach wizualizacji.

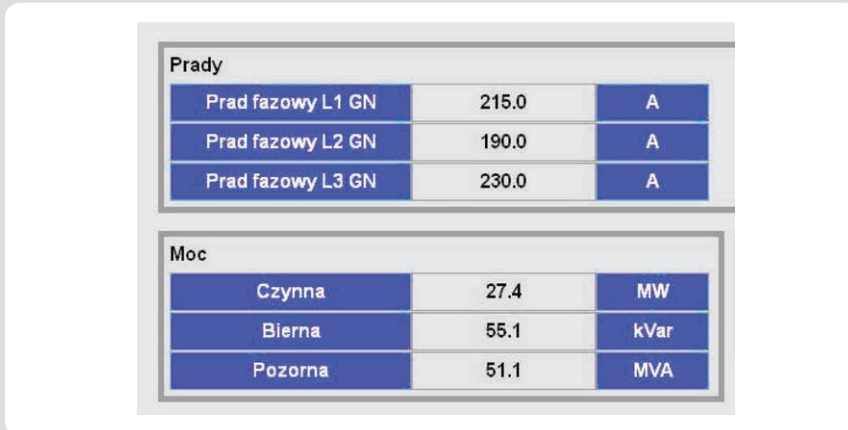
W programie symulującym pracę miernika prądu strony górnej transformatora zostały wprowadzone wartości: 215 A, 190 A i 230 A, zgodnie z rys. 9. Program wystawiał wprowadzone wartości pod określone adresy w protokole Modbus, zgodnie z dokumentacją rzeczywistego urządzenia. Zadaniem sterownika było automatyczne nawiązanie komunikacji z serwerem Modbus, pobranie wartości, przeskalowanie i zapis do bazy danych.



Rys. 9. Wygląd programu symulującego pracę serwera Modbus



Rys. 10. Fragment programu symulującego pracę klienta IEC 61850



Rys. 11. Fragment strony WWW pokazującej prądy oraz moc

Serwer stron WWW, pobierając w określonych interwałach czasowych dane z bazy, miał za zadanie ich wizualizację w odpowiednim panelu. Fragment strony pokazującej pobrane dane został przedstawiony

na rys. 11. Dodatkowo, w przypadku wykrycia przekroczenia dopuszczalnych wartości granic, wygenerowany zostałby komunikat ostrzegawczy, widoczny na panelu głównym wizualizacji.

Uruchomiony w oddzielnym wątku serwer IEC miał za zadanie pobranie zapisanych w bazie zmiennych i wystawienie ich do odczytu dla systemów SCADA. Fragment programu, pokazujący pobrane przez klienta IEC dane dotyczące wartości prądów po stronie górnej transformatora, został przedstawiony na rys. 10.

Oprócz programu IEC Browser, symulującego pracę systemów SCADA, do testów został wykorzystany klient Tamaracka oraz IEC Scout. Wszystkie programy pomyślnie przeszły testy kompatybilności z zaimplementowanym na sterowniku serwerem IEC. Jak widać na załączonych rysunkach, symulowane w serwerze Modbus wartości były prawidłowo odczytywane zarówno w samej wizualizacji na stronie WWW, jak i w programie pełniącym rolę klienta IEC 61850.

Bibliografia

1. Kowalik R., Kopański P., Glik K., „Autotransformer monitoring system” International Conference on Renewable Energies And Power Quality (ICREPQ '10), 23–25 March 2010, Grenada, Hiszpania, s. 692.
2. Kowalik R., Januszewski M., The example results of electrical power system protection devices interoperability tests using IEC 61850 standard, *Electrical Review* 2008, No. 9, s. 114–119.
3. Kowalik R., Januszewski M., Coordination of protections through the exchange of binary signals between power system substations in the IEC 61850 protocol, *Electrical Review* 2009, No. 6, s. 195–201.

Paweł Kopański

mgr inż.
Politechnika Warszawska | Twelve Electric sp. z o.o.
e-mail: pawel.kopanski@ien.pw.edu.pl

Ukończył studia na Wydziale Elektrycznym w Zakładzie Automatyki Elektroenergetycznej Politechniki Warszawskiej (2009). Od kilku lat uczestniczy w opracowywaniu oraz wdrażaniu systemów informatycznych. W głównej mierze są to systemy pomiarowe służące do koncentracji, agregacji oraz archiwizacji danych. Obecnie jest kierownikiem działu informatycznego w firmie Twelve Electric sp. z o.o.

Désiré D. Rasolomampionona

dr hab. inż. prof. PW
Politechnika Warszawska
e-mail: desire.rasolomampionona@ien.pw.edu.pl

Od 1994 roku pracuje na Wydziale Elektrycznym Politechniki Warszawskiej w Instytucie Elektroenergetyki. Obecnie jest kierownikiem Zakładu Aparatów i Automatyki Elektroenergetycznej. Jego zainteresowania naukowe koncentrują się głównie wokół problemów dotyczących automatyki elektroenergetycznej, sterowania pracą systemu elektroenergetycznego oraz zastosowania telekomunikacji i nowoczesnych technik informatycznych w elektroenergetyce.