Marcin BEDNAREK[1], Tadeusz DĄBROWSKI[2]

Politechnika Rzeszowska, Katedra Informatyki i Automatyki (1), Wojskowa Akademia Techniczna, Instytut Systemów Elektronicznych (2)

doi:10.15199/48.2019.11.43

Wybrane aspekty diagnozowania komunikacji w sieciach przemysłowych

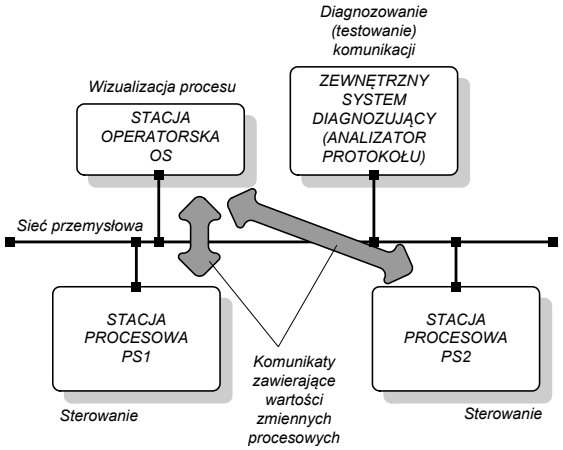
Streszczenie. *W artykule przedstawiono wybrane sposoby diagnozowania komunikacji w sieci przemysłowej. Rozpatruje się przypadek komunikacji pomiędzy stacjami rozproszonego mini-systemu sterowania: stacją operatorską i stacją procesową. Zakłada się, że stacje prowadzą komunikację pomiędzy sobą przy pomocy wybranego protokołu przemysłowego. Do testowania komunikacji używa się zwykle kosztownych i skomplikowanych obsługowo zestawów, złożonych z zewnętrznego urządzenia i dedykowanego oprogramowania. Proponuje się przeniesienie funkcji diagnostycznych do stacji operatorskiej oraz zastąpienie zewnętrznych analizatorów protokołów, prostymi i znacznie tańszymi narzędziami programowymi.*

Abstract*. The article presents selected ways of diagnosing communication in an industrial network. The case of communication between the stations of the mini-distributed control system, operator station as well as process station is considered. It is assumed that stations communicate with each other using the selected industrial protocol. Usually, to test communication, costly and complex (in terms of their operation) sets consisting of an external device and dedicated software are used. It is proposed to transfer the diagnostic functions to the operator station and to replace the external protocol analysers with simple and much cheaper software tools. (**Selected aspects of diagnosing communication in industrial networks**).*

Słowa kluczowe: sieć przemysłowa, układ dozorująco-terapeutyczny, diagnozowanie, analizator protokołu.
Keywords: industrial network, supervising and therapeutic system, diagnosis, protocol analyser.

Wstęp

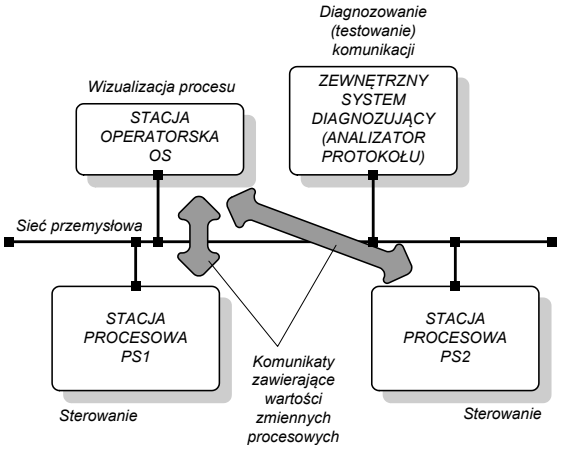
W artykule przedstawiono wybrane sposoby diagnozowania komunikacji w sieci przemysłowej. Rozpatruje się przypadek komunikacji (rys.1) pomiędzy stacjami rozproszonego mini-systemu sterowania (mini-DCS, ang. DCS – Distributed Control System): stacją operatorską (wizualizacja i oddziaływania operatorskie) i stacją procesową (sterowanie procesem). Zakłada się, że stacje prowadzą komunikację pomiędzy sobą przy pomocy wybranego protokołu przemysłowego. Ważnym elementem jest diagnozowanie komunikacji i utrzymanie zdatności systemu transmisji. Niezawodna komunikacja determinuje zdatność całego systemu sterowania. Do testowania komunikacji używa się zwykle kosztownych i skomplikowanych obsługowo zestawów, złożonych z zewnętrznego urządzenia i dedykowanego oprogramowania.



Rys.1. Diagnozowanie komunikacji z użyciem zewnętrznego systemu diagnozującego

Proponuje się przeniesienie funkcji diagnostycznych do stacji operatorskiej oraz zastąpienie zewnętrznych analizatorów protokołów, prostymi i znacznie tańszymi narzędziami programowymi (rys.2). Przedstawione w referacie rozwiązania dotyczą m.in.:

- przechwycenia i analizy treści komunikatów przesyłanych magistralą komunikacyjną przez oprogramowanie monitora portu szeregowego z wykorzystaniem budżetowego rozwiązania konwertera standardu RS485/RS232;
- możliwości wykorzystania pakietu wizualizacji do diagnozowania komunikacji prowadzonej w sieci Ethernet;
- implementacji funkcji testujących w języku skryptowym stacji operatorskiej, pozwalających na manualny wybór testów komunikacji.



Rys.2. Diagnozowanie komunikacji z wykorzystaniem stacji operatorsko-diagnostycznej

Diagnozowanie komunikacji Profibus

Rozpatrywana jest komunikacja wg protokołu Profibus prowadzona przy pomocy systemu magistralowego w standardzie RS485 [1] Do diagnozowania komunikacji prowadzonej wg protokołu Profibus [2] zazwyczaj stosuje się drogie analizatory protokołu. Przegląd rozwiązań można znaleźć w [3] Przykładem może być system sprzętowo-programowy firmy Procentec [4], składający się z interfejsu dołączanego poprzez wtyczkę z przelotką oraz oprogramowania diagnostycznego. Wspomniany interfejs sprzętowy (rys. 3a) podłącza się do komputera za pomocą przewodu USB. Dołączone dedykowane programowanie umożliwia testowanie połączeń pomiędzy stacjami komunikującego się systemu z dużą szczegółowością w pełnym zakresie prędkości magistrali komunikacyjnej. Można zadać pytanie: czy konieczne jest stosowanie drogiego analizatora? Czy nie można w jakiś sprytny sposób „podglądnąć" komunikacji stosując budżetowe

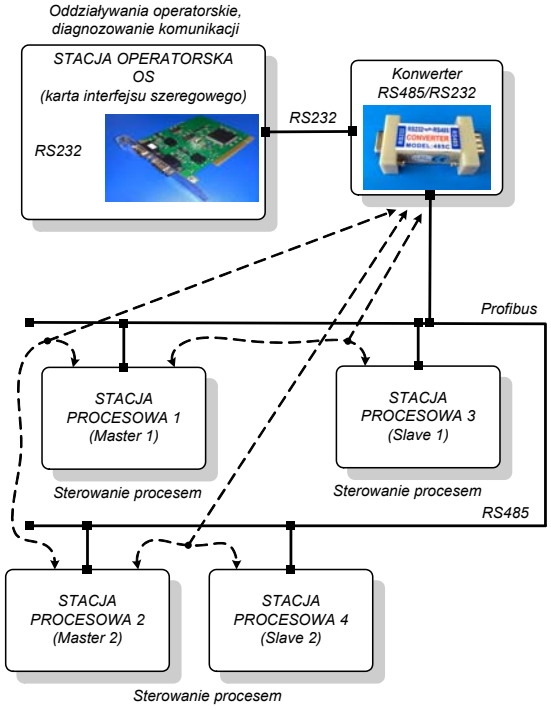
rozwiązania? Aby odpowiedzieć na to pytanie posłużono się układem testowym z rysunku 4.

a)

b)

Rys.3. Narzędzia do diagnozowania: a) profesjonalny interfejs analizatora Profibus firmy Procentec [4]; b) pasywny konwerter standardu RS485/RS232

Układ testowy (rys. 4) składa się z dwóch stacji nadrzędnych master oraz dwóch urządzeń podrzędnych slave. Komunikacja wg Profibus prowadzona jest wg modelu multimaster [1]. Oznacza to, że wśród komunikatów przesyłanych magistralą komunikacyjną znajdują się komunikaty-polecenia pochodzące od stacji nadrzędnych, ale także komunikaty z wymian master-master.



Rys.4. Diagnozowanie komunikacji szeregowej

Ze względu na stosowany model wymiany danych, urządzenia nadrzędne muszą wymieniać token (specjalny komunikat). Tylko urządzenie, dysponujące w danej chwili tokenem, może nadawać, odpytując tym samym swoje slave'y. W celu dozorowania komunikacji wykorzystano właściwość magistrali szeregowej, która pozwala na dołączenie do niej do wielu odbiorników. Przygotowania do dozorowania komunikacji objęły:

- obniżenie prędkości przesyłu danych na magistrali do 9600 b/s;
- zastosowanie układu budżetowego konwertera sprzętowego RS485/RS232 (rys. 3b);
- wyposażenie stacji operatorskiej w kartę interfejsu RS232;
- zainstalowanie na stacji operatorskiej dowolnej aplikacji monitora portu szeregowego, np. [5];
- połączenie całości wg schematu z rys. 4.

Diagnozowanie komunikacji w omawianym przykładzie polega na sprawdzeniu, czy urządzenia master prawidłowo przekazują komunikat tokena. W tym celu zbadano 3 przypadki. Obserwowano ruch sieciowy na magistrali komunikacyjnej w konfiguracji:

(I) włączonej stacji procesowej nr 1, wyłączonej stacji procesowej nr 2;
(II) włączonej stacji procesowej nr 2, wyłączonej stacji procesowej nr 1;
(III) włączonych obydwu stacji procesowych nr 1 i nr 2.

W każdym z przypadków, stan pozostałych urządzeń slave nie ma znaczenia, ponieważ nie szukano wśród transmitowanych komunikatów ramek z danymi, lecz tylko tych, które zawierają informację o przekazaniu tokena. Dla uproszczenia rozważań przyjęto, że z punktu widzenia diagnosty, pozostałe przechwycone ramki nie są „interesujące".

```
...20 8 0 0 C 16 DC 1 1 68 5 5 68 20 1 7D 0 0 81
16 68 5 5 68 1 20 8 0 0 C 16 10 13 1 49 5D 16 68
5 5 68 20 1 5D 0 0 61 16 68 5 5 68 1 20 8 0 0 C
16 DC 1 1 68 5 5 68 20 1 7D 0 0 81 16 68 5 5 68
1 20 8 0 0 C 16 10 14 1 49 5E 16 68 5 5 68 20 1
5D 0 0 61 16 68 5 5 68 1 20 8 0 0 C 16 DC 1 1 68
5 5 68 20 1 7D 0 0 81 16 68 5 5 68 1 20 8 0 0 C
16 10 15 1 49 5F 16 68 5 5 68 20 1 5D 0 0 61 16
68 5 5 68 1 20 8 0 0 C 16 DC 1 1 68 5 5 68 20 1
7D 0 0 81 16 68 5 5 68 1 20 8 0 0 C 16 10 16 1
49 60 16 68 5 5 68 20 1 5D 0 0 61 16 68 5 5 68 1
20 8 0 0 C 16 DC 1 1 68 5 5 68 20 1 7D...
```

Rys.5. Diagnozowanie komunikacji – przypadek (I)

W wariancie (I) należy zauważyć (rys. 5), że wśród przechwyconych komunikatów można odnaleźć prawidłowo transmitowany komunikat przekazania znacznika, oznaczony na rysunku DC 1 1 . Ze względu na to, iż tylko jeden master jest aktualnie włączony, ramka przekazania znacznika rozpoczynająca się znacznikiem DC jest formalnym przekazaniem tokena urządzenia „do siebie" (od „numeru jeden" do „numeru jeden").

```
...10 5 2 49 50 16 10 2 5 0 7 16 10 5 2 5D 64 16
68 5 5 68 2 5 8 0 0 F 16 DC 2 2 10 0 2 49 4B 16
10 5 2 7D 84 16 68 5 5 68 2 5 8 0 0 F 16 DC 2 2
10 1 2 49 4C 16 10 5 2 5D 64 16 68 5 5 68 2 5 8
0 0 F 16 DC 2 2 10 3 2 49 4E 16 10 5 2 7D 84 16
68 5 5 68 2 5 8 0 0 F 16 DC 2 2 10 4 2 49 4F 16
10 5 2 5D 64 16 68 5 5 68 2 5 8 0 0 F 16 DC 2 2
10 5 2 49 50 16 10 2 5 0 7 16 10 5 2 7D 84 16 68
5 5 68 2 5 8 0 0 F 16 DC 2 2 10 0 2 49 4B 16 10
5 2 5D 64 16 68 5...
```

Rys.6. Diagnozowanie komunikacji – przypadek (II)

W wariancie (II), podobnie jak w (I), należy zauważyć (rys. 6), że wśród przechwyconych komunikatów można odnaleźć prawidłowo transmitowany komunikat przekazania znacznika, oznaczony na rysunku DC 2 2 . Tym razem także, z powodu braku urządzenia nr 1 (wyłączone) ramka przekazania znacznika rozpoczynająca się znacznikiem DC jest formalnym przekazaniem tokena - od urządzenia nr 2 do nr 2.

W wariancie (III) potwierdzeniem prawidłowej wymiany tokenów pomiędzy stacjami powinny być wzajemne wymiany komunikatów.

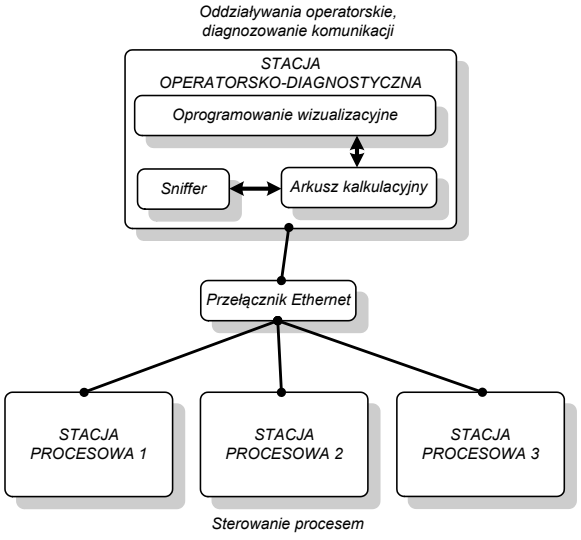
```
...10 6 2 49 51 16 10 5 2 5D 64 16 68 5 5 68 2 5
8 0 0 F 16 DC 1 2 68 5 5 68 14 1 7D 0 0 92 16 68
5 5 68 1 14 8 0 0 1D 16 DC 2 1 10 7 2 49 52 16
10 5 2 7D 84 16 68 5 5 68 2 5 8 0 0 F 16 DC 1 2
68 7 7 68 FF 81 46 3A 3E 0 0 3E 16 68 5 5 68 14
1 5D 0 0 72 16 68 5 5 68 1 14 8 0 0 1D 16 68 7 7
68 FF 81 46 3A 3E 0 0 3E 16 68 5 5 68 14 1 7D 0
0 92 16 68 5 5 68 1 14 8 0 0 1D 16 68 7 7 68 FF
81 46 3A 3E 0 0 3E 16 68 5 5 68 14 1 5D 0 0 72
16 68 5 5 68 1 14 8 0 0 1D 16 68 7 7 68 FF 81 46
3A 3E 0 0 3E 16 68 5 5 68 14 1 7D 0 0 92 16 68 5
5  68 1 14 8 0 0 1D 16...
```

Rys.7. Diagnozowanie komunikacji – przypadek (III)

Analiza ruchu sieciowego przechwyconego dla wariantu (III) (rys. 7) potwierdza prawidłowe zachowanie urządzeń master, które przekazują naprzemiennie token (zaznaczone na rys. 7 kolejno występujące komunikaty ...DC 1 2, DC 2 1, DC 1 2...). Przytoczony przykład pokazuje, że zastosowane proste i tanie rozwiązanie pozwala także na przeprowadzenie dozorowania komunikacji prowadzonej wg standardu Profibus. Oczywiście nie należy zapominać o występujących ograniczeniach (niewielkiej prędkości, braku filtrów, żmudnego procesu rozpoznawania ramek).

Diagnozowanie komunikacji Ethernet

Diagnozowanie prowadzone jest za pomocą stacji operatorskiej dołączonej do sieci Ethernet [7]. Takim sposobem diagnozować można np. komunikację wg Profinet lub Modbus TCP. Zakłada się wykorzystanie do analizy oprogramowania wywoływanego cyklicznie *sniffera* [8], arkusza kalkulacyjnego oraz pakietu wizualizacji [9]. Rysunek 8 przedstawia schematycznie elementy układu prowadzącego dozorowanie komunikacji.



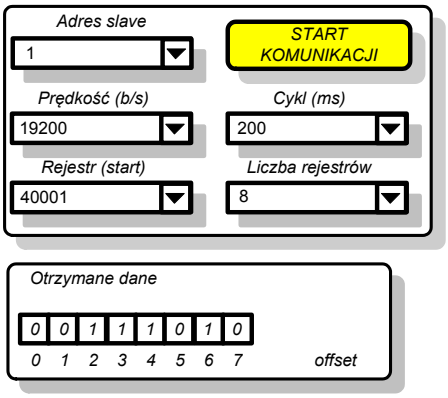
Rys.8. Diagnozowanie komunikacji Ethernet

Załóżmy, dla uproszczenia, że celem dozorowania jest znalezienie przesyłanej wartości zmiennej procesowej. Diagnosta zna położenie danych znaczących w transmitowanym komunikacie. Za pomocą cyklicznie uruchamianego konsolowego *sniffera* tworzone są pliki (o rozszerzeniu *.csv*) z zawartością kilkusekundowego ruchu sieciowego. Jeśli diagnozowano by komunikację pomiędzy innymi stacjami niż operatorsko-diagnostyczna, należałoby dokonać odpowiednich czynności powodujących, że komunikaty pomiędzy obserwowanymi stacjami trafiałyby także do portu, do którego przyłączona jest stacja operatorsko-diagnostyczna. Cykliczność uruchamiania zapisu ruchu sieciowego zapewnia napisany skrypt lub zwykłe zadanie harmonogramu zadań cyklicznych systemu operacyjnego. Następnie otwierany jest, także wg harmonogramu, arkusz kalkulacyjny wraz z zapisanym wcześniej plikiem *.csv*. Kolejnym krokiem jest uruchomienie pakietu wizualizacji, który odwołuje się do komórek arkusza kalkulacyjnego zawierającego zgromadzone dane. Oprogramowanie wizualizacyjne, przy pomocy wewnętrznych skryptów umożliwia wyszukanie odpowiednich danych wśród komórek arkusza. Proces kończy się wygenerowaniem alarmu dla operatora dotyczącym procesu dozorowania wartości zmiennej. Tym samym możliwe jest potwierdzenie lub zaprzeczenie, że przesyłana zmienna procesowa przyjmuje założoną wartość.

Elementy testowania komunikacji Modbus

W przypadku, gdy operatorowi zależy na wygenerowaniu szeregu testów, sprawdzających poprawność zachowania się urządzeń komunikujących się według protokołu Modbus [10], zamiast skomplikowanego analizatora protokołu, może w tym celu wykorzystać także gotowy pakiet wizualizacji. Protokół Modbus bazuje na modelu wymiany danych master-slave. Urządzenie nadrzędne (master) odpytuje urządzenia podrzędne (slave). Komunikacja ma charakter „pytanie-odpowiedź”. Więcej informacji na temat protokołu Modbus można znaleźć w [9-11]. Często zachodzi potrzeba wygenerowania testów, celem których jest:

- odpytanie urządzenia podrzędnego o zadanym adresie o wartość pewnej zmiennej;
- odpytanie urządzenia podrzędnego o wartość pewnej zmiennej z zadanym czasem cyklu;
- odpytanie urządzenia podrzędnego o zadaną liczbę wartości zmiennych;
- odpytanie urządzenia podrzędnego z zadaną prędkością przesyłu.



Rys.9. Testowanie komunikacji Modbus

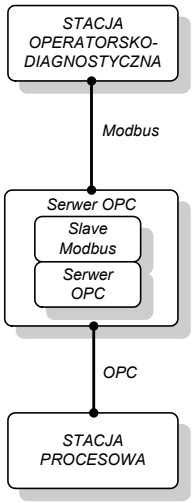
Do tego typu testów, wygodnie jest użyć oprogramowania wizualizacyjnego stacji operatorskiej, które umożliwi operatorowi skonfigurowanie, a następnie uruchomienie komunikacji z zadanymi parametrami transmisji. Jako przykład posłuży tu wspomniany wcześniej

pakiet wizualizacji Intouch [9]. Dzięki użyciu pól zmiennych [12,13], które przechowują dodatkowe parametry komunikacyjne zmiennych wejściowych/wyjściowych oraz graficznych elementów rozwijanych menu, można utworzyć pulpit konfiguracyjny do testowania komunikacji wg Modbus. Schematyczną strukturę takiego pulpitu konfiguracyjnego przedstawia rysunek 9. Skonfigurowany na rysunku 9 test diagnostyczny, powoduje otrzymanie odpowiedzi od urządzenia slave o adresie 1, dotyczącej zawartości kolejnych ośmiu rejestrów binarnych od adresu 40001 z czasem cyklu (aktualizacji) 200ms, z prędkością 19200 b/s. Ramka „Otrzymane dane" pokazuje rezultat wykonania testu.

Wykonanie aplikacji umożliwiającej testowanie komunikacji w protokole Modbus, możliwe jest dzięki odpowiedniemu sposobowi konfiguracji pakietu wizualizacji. Można to opisać w kilku krokach:

1. Dla każdej zmiennej wejściowej lub wyjściowej definiowana jest zasób nazwa dostępu (*access name*).
2. W nazwie dostępu wskazuje się:
   a) nazwę aplikacji lub nazwę programowego drivera, używanego później do komunikacji;
   b) *temat (topic)* zawierający ustawienia komunikacyjne.
3. Konfiguruje się temat (w danym przypadku zdefiniowano wiele tematów), a w nim:
   a) numer portu komunikacyjnego;
   b) prędkość transmisji;
   c) adres urządzenia podrzędnego;
   d) liczbę kolejnych rejestrów do odczytu i (oddzielnie) do zapisu.

Wybierając z rozwijanych menu (rys. 9) odpowiednie parametry komunikacyjne, docelowo do zmiennej, której wartość chcemy uzyskać, przyporządkowana zostaje odpowiednio skonfigurowana nazwa dostępu zawierająca właściwy temat. Należy zaznaczyć, że wybór parametrów możliwy jest przy wykorzystaniu rozwijanych menu. Ogranicza to w pewnym stopniu, i tak ogromną, liczbę tematów niezbędnych do predefiniowania.



Rys.10. Połączenie Modbus - OPC

Podczas diagnozowania komunikacji prowadzonej według Modbus pojawia się niekiedy sytuacja braku obsługi protokołu przez niektóre stacje systemu. Można wtedy zastosować rozwiązanie, które będzie pośrednikiem pomiędzy stacją operatorsko-diagnostyczną, a testowaną stacją. Wtedy wygodnym rozwiązaniem jest zastosowanie protokołu OPC [15], który wspierany jest przez większość producentów oprogramowania przemysłowego. Realizacja dodatkowego połączenia mostowego w układzie przedstawionym na rysunku 10, pozwala na dostęp do zmiennych procesowych za pomocą protokołu Modbus. Serwer OPC jest jednocześnie, od strony stacji diagnostyczno-operatorskiej urządzeniem slave, które można odpytać o odpowiednie wartości zmiennych procesowych.

Podsumowanie

Przeniesienie funkcji diagnostycznych do stacji operatorskiej oraz implementacja ww. wybranych aspektów diagnozowania komunikacji w sieci przemysłowej pozwala na:

- znaczne oszczędności finansowe;
- wygodne umieszczenie funkcji diagnostycznych bezpośrednio w stacji operatorskiej;
- uproszczenie procesu testowania;
- realizację alarmowania w pakiecie SCADA (w przypadku pojawiających się błędów).

Zaproponowane wykorzystanie serwera OPC do diagnozowania zmiennych procesowych transmitowanych w systemie daje dodatkowe korzyści. Odpowiednie połączenie (linkowanie) zmiennych w serwerze OPC może umożliwić uzyskanie pomocniczych informacji, np. dotyczących stempla czasowego otrzymanej wartości zmiennej.

***Autorzy**: prof. dr hab. inż. Tadeusz Dąbrowski, Wojskowa Akademia Techniczna, Instytut Systemów Elektronicznych, ul. Kaliskiego 2, 01-476 Warszawa, E-mail: tdabrowski@wat.edu.pl; dr inż. Marcin Bednarek, Politechnika Rzeszowska, Katedra Informatyki i Automatyki, ul. Skłodowskiej 8, 35-959 Rzeszów, E-mail: bednarek@prz.edu.pl.*

LITERATURA

[1] https://elektronikab2b.pl/technika/3404-czym-jest-rs-485, [odczyt w dniu 2019-04-20]
[2] Intex sp. z o.o., Wprowadzenie do diagnostyki sieci Profibus DP oraz PA, *IV Konferencja Organizacji PROFIBUS PNO Polska*, Tomaszowice 9-10 styczeń 2008
[3] Mossin E. A., Brandão D., Intelligent diagnostic for PROFIBUS DP networks, *2012 IEEE International Conference on Industrial Technology*, Athens, 2012, 772-777
[4] Strona domowa Procentec, https://procentec.com/, [odczyt w dniu 2019-04-20]
[5] https://strokescribe.com/en/read-serial-port-excel-2007.html, [odczyt w dniu 2019-04-20]
[6] Strona pobierania firmy Commfront, https://www.commfront.com/pages/downloads, [odczyt w dniu 2019-04-20]
[7] Stępień J., Kołodziej J., Dziurdzia P., Machowski W., Golański R., Precise Time Distribution and Time Synchronized Transmission Aspects in the Industrial Ethernet Networks, *Przegląd Elektrotechniczny*, 89 (2013), nr 12, 37-40
[8] https://www.wireshark.org/docs/man-pages/tshark.html, [odczyt w dniu 2019-04-20]
[9] Dokumentacja elektroniczna pakietu Wonderware Intouch.
[10] MODBUS over Serial Line Specification and Implementation Guide, Modbus.org, 2005, [http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf]
[11] Carcano A., Fovino I.N., Masera M., Trombetta A., State-Based Network Intrusion Detection Systems for SCADA Protocols: A Proof of Concept In Rome E., Bloomfield R. (eds.), *Critical Information Infrastructures Security*, Springer-Verlag, Berlin-Heidelberg 2010, 138-150
[12] Układ dozorująco-terapeutyczny systemu transmisji danych w sieci przemysłowej, *Konferencja Diag 2019*, Augustów, 20-25 maja 2019
[13] Wonderware FactorySuite InTouch, Opis funkcji, pól i zmiennych systemowych, Invensys Systems, Inc., 2005
[14] Kwiecień R., Szychta L., Figura R., Skryptowy informatyczny system sterowania urządzeniami automatyki przemysłowej, *Przegląd Elektrotechniczny*, 86 (2010), nr 2, 285 - 288
[15] Kwiecień R., Szychta E., Szychta L., Data acquisition in OPC-based industrial IT systems, *The 4th international conference on electrical and control technologies*, ECT 2009