Diagnozowanie komunikacji między elementami rozproszonego systemu sterowania

Marcin Bednarek
Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki, Katedra Informatyki i Automatyki, al. Powstańców Warszawy 12, 35-959 Rzeszów

Streszczenie: W artykule opisano wybrane fragmenty procesu diagnozowania komunikacji między stacją procesową i operatorską, a także między stacjami procesowymi minisystemu rozproszonego, zbudowanego na bazie modułowego sterownika przemysłowego AC800F. W wyniku przeprowadzonych eksperymentów uzyskano informacje dotyczące sposobu transmisji i położenia wartości zmiennych procesowych w przesyłanych komunikatach. Informacje te można wykorzystać do podjęcia decyzji dotyczących dodatkowych zabezpieczeń przesyłu lub skomunikowania stacji systemu z rozszerzającymi zasobami użytkownika.

Słowa kluczowe: sieci przemysłowe, rozproszony system sterowania, stacja procesowa, stacja operatorska, diagnozowanie komunikacji, przesył danych

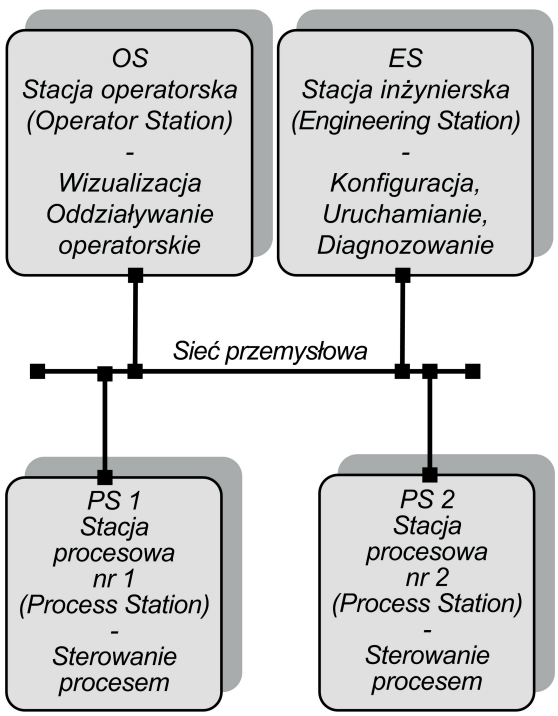
1. Wprowadzenie

W skład rozproszonych systemów sterowania DCS (ang. *Distributed Control System*) wchodzi kilka rodzajów stacji [1, 2]. Są to głównie:
- stacje procesowe PS (ang. *Process Station*) – sterowniki przemysłowe odpowiedzialne za realizację programu sterowania procesem przemysłowym;
- stacje operatorskie OS (ang. *Operator Station*) – pełnią rolę systemów wizualizacji, oddziaływania operatorskiego oraz obsługi alarmów;
- stacje inżynierskie ES (ang. *Engineering Station*) – zwykle komputery przenośne, służące do konfiguracji pozostałych stacji systemu, uruchamiania i diagnozowania działania systemu oraz komunikacji [3].

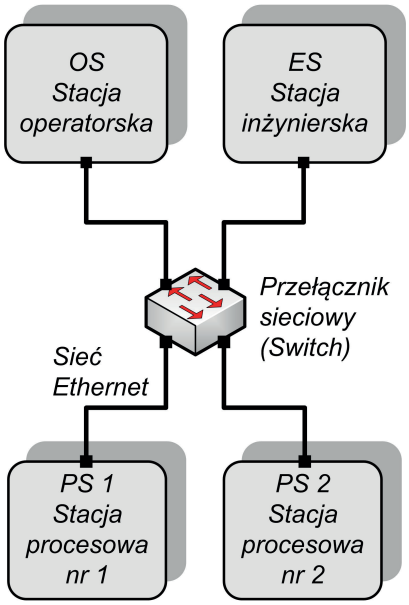
W jednym z wariantów (Rys. 1), stacje systemu mogą być połączone siecią przemysłową bazującą na architekturze magistralowej [4]. Dane przesyłane magistralą trafiają do wszystkich podłączonych urządzeń. Ze względu na zasadniczą cechę sieci przemysłowej, którą powinna się charakteryzować, czyli determinizm czasowy dostarczania wiadomości – zastosowany protokół komunikacyjny powinien wprowadzać odpowiedni mechanizm dostępu do łącza. Powinien on sterować w odpowiedni sposób interfejsami nadawczymi stacji systemu zapobiegając ewentualnym kolizjom. Typowymi przedstawicielami tego rodzaju sieci są Modbus [5, 6] oraz Profibus [7, 8]. W każdym z wymienionych standardów stosowany jest mechanizm odpytywania (ang. *polling*) urządzeń podrzędnych przez nadrzędne (tzw. *master-slave*). Dodatkowo Profibus oferuje możliwość korzystania z wielu masterów, dlatego zastosowano w nim mechanizm wymiany żetonu (ang. *token*). Tylko urządzenie aktualnie wyposażone w *token* może nadawać. W obydwu standardach doskonale znane są pola danych komunikatów i miejsca przesyłu wartości zmiennych w ramce. Diagnozowanie komunikacji w tym wariancie nie wymaga dodatkowych zabiegów, wystarczy bowiem zainstalowanie, w jednej ze stacji, oprogramowania monitorującego bezproblemowo wszystkie

Autor korespondujący:
Marcin Bednarek, bednarek@prz.edu.pl

Artykuł recenzowany
nadesłany 18.11.2022 r., przyjęty do druku 12.12.2022 r.



Rys. 1. System wykorzystujący magistralową sieć polową
Fig. 1. System using a fieldbus network



Rys. 2. Rozpatrywany system wykorzystujący standard Ethernet
Fig. 2. System under consideration using the Ethernet standard

komunikaty wysyłane na magistralę. Chcąc rozbudować system o nowe elementy wystarczy oprogramować interfejs komunikacyjny zgodnie z podanymi standardami [5–8].

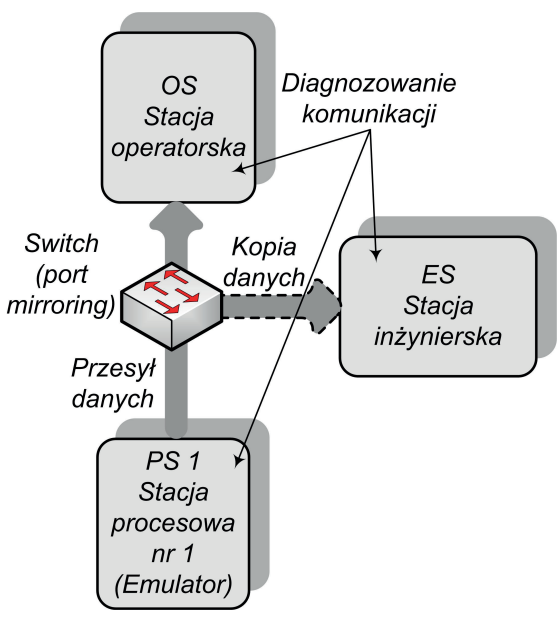
Innym sposobem połączenia urządzeń rozproszonego systemu sterowania jest wykorzystanie struktury gwiaździstej (Rys. 2) i standardu Ethernet [9, 10]. Elementy systemu połączone są za pomocą przełącznika sieciowego [11]. Do przełącznika podłączona jest także diagnozująca stacja inżynierska. W celu poprawnego diagnozowania komunikacji, oprócz zainstalowania odpowiedniego oprogramowania *sniffera* umożliwiającego przechwyt danych (co wystarczyło w przypadku z Rys. 1), należy skierować do niego „kopię" ruchu sieciowego. Jednym z rozwiązań jest odpowiednie ustawienie przełącznika, stosując np. port lustrzany [11, 12], na który dodatkowo jest przesyłany cały ruch sieciowy pomiędzy diagnozowanymi stacjami. Stację diagnozującą podłącza się wtedy do przygotowanego w ten sposób portu przełącznika. Takie połączenie występuje w systemie rozpatrywanym w dalszej części artykułu. Istnieją dwie tendencje w realizacji komunikacji między elementami systemu. Jedną z nich jest wykorzystanie standardowych rozwiązań bazujących na sieci Ethernet. Należy wspomnieć tu o najbardziej popularnych „następcach" Modbus i Profibus, czyli Modbus TCP [13, 14] lub też Profinet [15, 23, 24]. W tym przypadku również znana jest budowa pola danych pakietu TCP lub datagramu UDP [16]. Jednak niektórzy producenci decydują się na zastosowane do wymiany danych między stacją procesową, operatorską i inżynierską własnych, zamkniętych, nieudokumentowanych protokołów komunikacyjnych [25].

Diagnozowanie nieudokumentowanych, firmowych protokołów przeprowadza się na zasadzie badania „czarnej skrzynki" [17]. Oznacza to, że podczas badania dysponuje się jedynie znajomością wartości danych wysyłanych i odbieranych. Natomiast nie są znane informacje dotyczące m.in. rozmieszczenia zmiennych w polu danych komunikatu lub sposobu ich przesyłania. Strukturę pola danych można poznać w wyniku diagnozowania komunikacji. Ze względu na to, że diagnozowanie dotyczy firmowego, zamkniętego protokołu [25], nie został on dotychczas przedstawiony w ogólnodostępnych źródłach. Istnieją wprawdzie informacje dotyczące komunikacji w podobnym systemie [26], jednak ograniczają się do przedstawienia znanych typów komunikacji. Badania komunikacji koncentrują się wokół testowania podatności systemów DCS na ataki [20, 27] i luk w zabezpieczeniach [27]. Diagnozowanie komunikacji obejmuje też zagadnienia zabezpieczeń zewnętrznych, np. stosowania firewalli [18, 28], czy też szyfrowania danych i dystrybucji kluczy [19, 29]. Badana jest również odporność systemów na ataki typu DDoS [13]. W większości jednak badania poświęcone są testowaniu udokumentowanych protokołów przemysłowych, w tym protokołu MMS [20] w sterownikach AC800M. Dedykowane, zamknięte, firmowe protokoły w dalszym ciągu wymagają poznania ich parametrów i funkcjonalności.

W rozpatrywanym w artykule systemie, zbudowanym na bazie stacji procesowej AC800F i oprogramowania wizualizacyjnego DigiVis wykorzystano właśnie tego rodzaju firmowe, dedykowane rozwiązanie [25]. Czy takie rozwiązanie jest lepsze? Można pozornie odnieść wrażenie, że przecież zamknięty, dedykowany protokół komunikacyjny wydaje się bezpieczniejszy, ze względu na jego słabą znajomość przez ewentualnych intruzów, próbujących dostać się do systemu i zaburzyć jego prawidłowe funkcjonowanie. Czy należy czuć się komfortowo i bezpiecznie przesyłając dane ze stacji procesowej do operatorskiej, wizualizującej operatorowi trwający, sterowany proces przemysłowy właśnie takim firmowym, zamkniętym protokołem? Ze względu na ogólną, prawidłową tendencję do tworzenia sieci przemysłowej jako oddzielnej izolowanej struktury, można by odpowiedzieć twierdząco. Z drugiej strony, informacje przedstawione w części 2, dotyczące diagnozowania komunikacji, mogą nieco ostudzić entuzjazm zwolenników zamkniętych rozwiązań.

2. Diagnozowanie komunikacji PS-OS

Problem diagnozowania komunikacji stacja procesowa – stacja operatorska (PS-OS), jak każdy proces diagnozowania składa się z dwu etapów: badania diagnostycznego i wnioskowania [18]. Wnioskowanie przeprowadzono na podstawie danych uzyskanych w wyniku badania diagnostycznego zrealizowanego w układzie przedstawionym na Rys. 3. Stacja procesowa połączona jest z operatorską przez przełącznik sieciowy. Do przełącznika podłączona jest także diagnozująca stacja inżynierska z zainstalowanym oprogramowaniem *sniffera* przechwytującym transmitowane dane. Zastosowano tu także odpowiednie ustawienie przełącznika (tzw. port lustrzany [11]), dołączając stację diagnozującą do portu, na który kopiowany jest cały ruch pomiędzy stacjami. Dodatkowym utrudnieniem w danym przy-



Rys. 3. Diagnozowanie komunikacji pomiędzy stacjami operatorskimi
Fig. 3. Diagnosing communication between operator stations

padku jest brak znajomości struktury komunikatu, co zmusza do wielokrotnego powtarzania przechwytywania transmitowanych danych, kolejnych zmian wartości przesyłanych danych i obserwacji zmiany wartości bajtów pola danych datagramu (stwierdzono po rozpoczęciu nasłuchu komunikacji, że stacje wymieniają dane wg protokołu UDP) [16]. Komunikacja wg UDP jest szybsza niż wg TCP [16]. Przesyłane komunikaty w przeciwieństwie do transmisji TCP nie muszą być potwierdzane przez stronę odbiorczą. Mimo szybkości oferowanej przez połączenie UDP, zastosowanie protokołu bezpołączeniowego (bez potwierdzania) do komunikacji ze stacją operatorską, gdzie wizualizuje się sterowany proces i przeprowadza się oddziaływanie operatorskie jest problematyczne ze względu na bezpieczeństwo dostarczenia danych.

W celu zdiagnozowania miejsca w komunikacie, gdzie przesyłane są wartości zmiennych procesowych, w stacji procesowej uruchomiono nieskomplikowany układ sterowania (Rys. 4), tzw. układ „start-stop" [22] w języku FBD [30]. Sygnały na wejściach i wyjściach są binarne. Realizuje on funkcję podtrzymania stanu wysokiego na wyjściu, w układzie dwóch wejść obsługiwanych z przycisków monostabilnych, z priorytetem wejścia „stop" – opisaną wzorem:

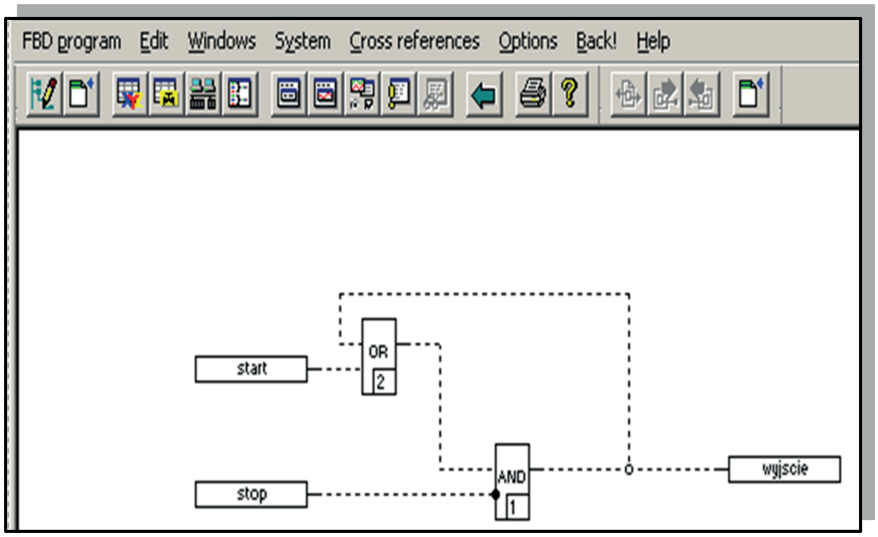
$$wyjscie_i = (start_i \vee wyjscie_{i-1}) \wedge \overline{stop}$$

gdzie: *start*, *stop* – zmienne wejściowe; *wyjscie* – zmienna wyjściowa; oznaczenia z indeksem dolnym (i–1) dotyczą wartości zmiennej z poprzedniego cyklu obliczeniowego, oznaczenia z indeksem (i) – cyklu aktualnego.

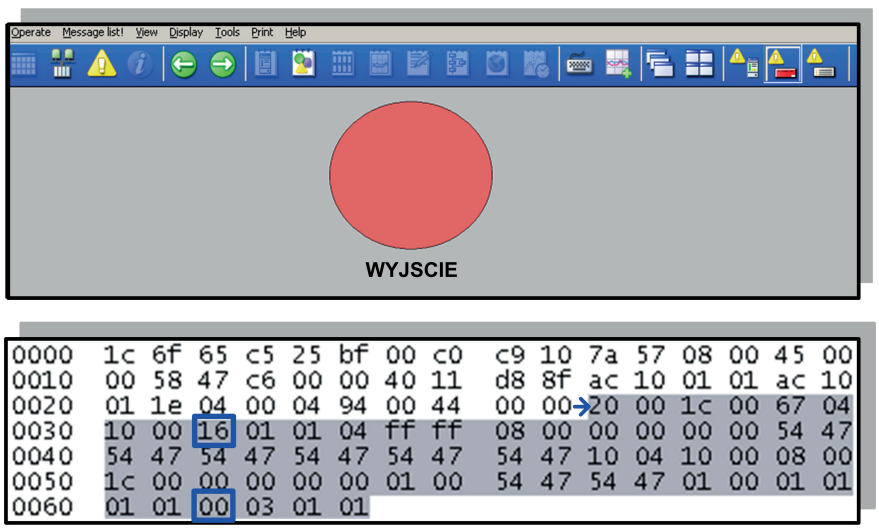
Ten prosty układ „strat-stop", po podłączeniu w strukturze sprzętowej do wejść i wyjścia sterownika, umożliwia obsługę wyjścia z poziomu fizycznych przycisków monostabilnych.

Na stacji operatorskiej uruchomiono nieskomplikowaną wizualizację, pokazującą stan zmiennej *wyjscie* za pomocą statycznego koła zmieniającego kolor (*wyjscie* = 0 – czerwony, *wyjscie* = 1 zielony). Zastosowanie jak najprostszego układu sterowania i wizualizacji miało za zadanie zminimalizować liczbę przesyłanych danych między stacjami, w celu uproszczenia wnioskowania diagnostycznego dotyczącego znaczenia pól w komunikacie.

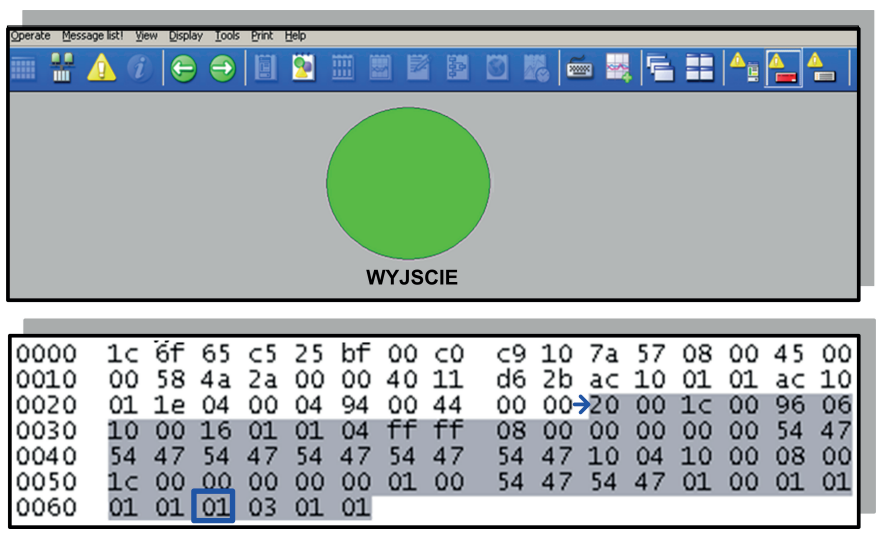
W pierwszej fazie eksperymentu pozostawiono wyjście w stanie 0. Na ekranie stacji operatorskiej uzyskano obraz koła w kolorze czerwonym (Rys. 5). Jednocześnie uruchomiony *sniffer* zarejestrował datagram, którego pole danych przedstawia rysunek 5. Niebieską ramką zaznaczono wartości pól, na które należy



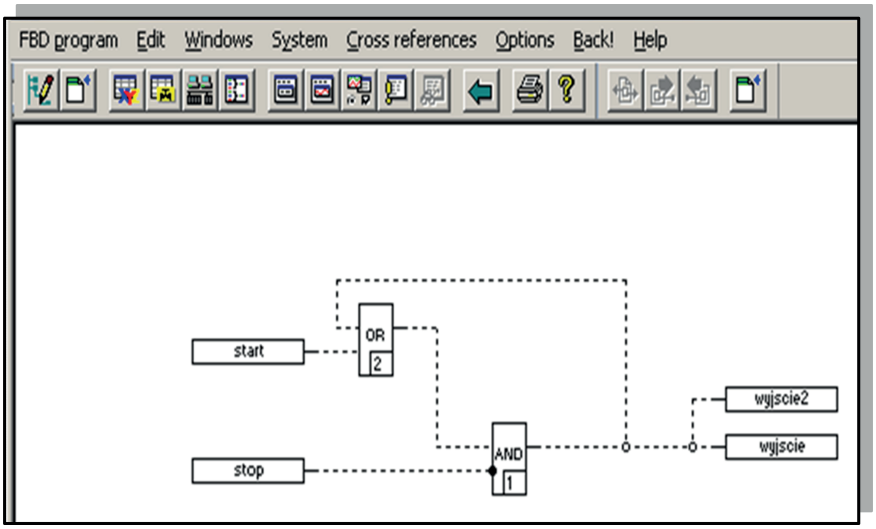
Rys. 4. Testowy schemat FBD – przesył jednej wartości (*wyjscie* = 0)
Fig. 4. FBD test diagram – one value transfer (*wyjscie* = 0)



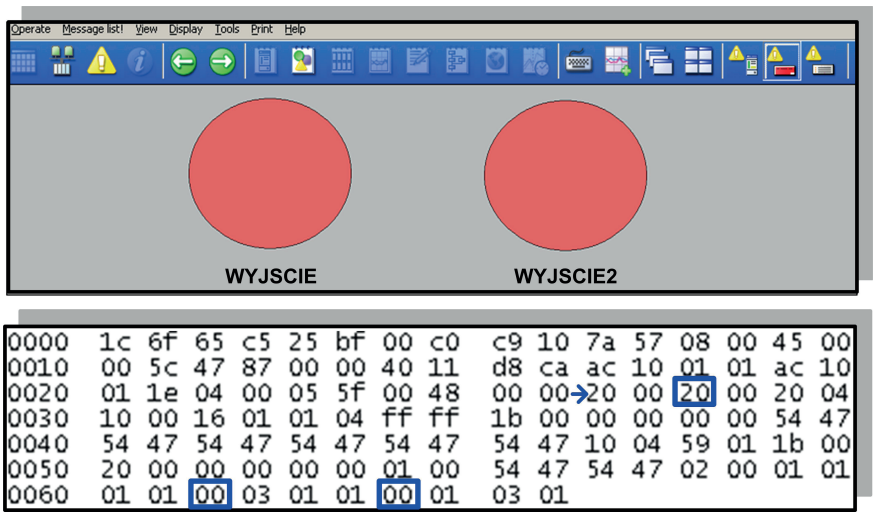
Rys. 5. Wizualizacja i przesyłane dane dla przypadku: *wyjscie* = 0
Fig. 5. Visualisation and data transferred for the case: wyjscie = 0



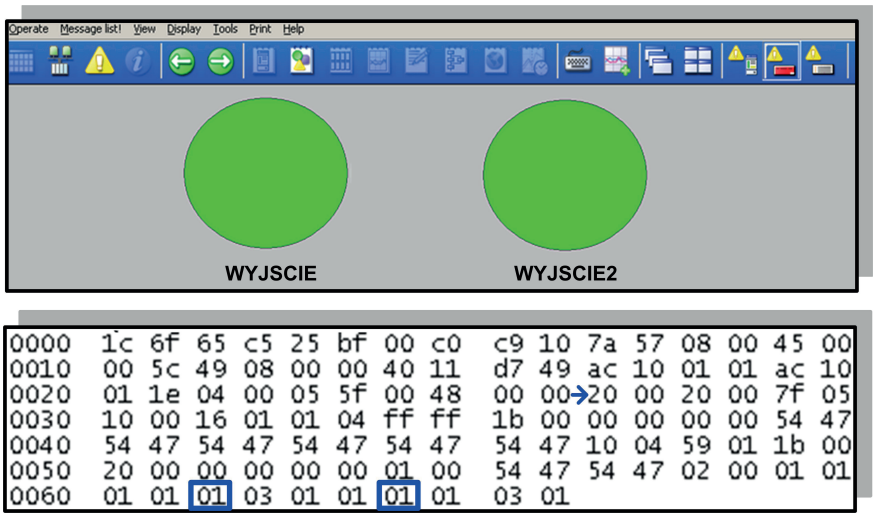
Rys. 6. Wizualizacja i przesyłane dane dla przypadku: *wyjscie* = 1
Fig. 6. Visualisation and data transferred for the case: *wyjscie* = 1



Rys. 7. Testowy schemat FBD – przesył dwóch wartości (*wyjscie* = 0, *wyjscie*2 = 0)
Fig. 7. FBD test diagram – two value transfer (*wyjscie* = 0, *wyjscie*2 = 0)



Rys. 8. Wizualizacja i przesyłane dane dla przypadku: *wyjscie* = 0, *wyjscie*2 = 0
Fig. 8. Visualisation and data transferred for the case: *wyjscie* = 0, *wyjscie*2 = 0



Rys. 9. Wizualizacja i przesyłane dane dla przypadku: *wyjscie* = 1, *wyjscie*2 = 1
Fig. 9. Visualisation and data transferred for the case: *wyjscie* = 1, *wyjscie*2 = 1

zwrócić uwagę. Zwrócono uwagę na wartość 16(hex), która zawarta była we wszystkich komunikatach i oznaczała numer systemowy stacji operatorskiej (dziesiętnie 22). Na czwartym bajcie od końca pola danych zarejestrowano wartość 00. Aby potwierdzić, że jest to wartość zmiennej *wyjscie*, należało zmienić jej stan i ponownie obserwować zawartość pola danych datagramu.

Wielokrotne próby przełączania przyciskami (*start*, *stop*) dołączonymi do wejść binarnych sterownika, wysterowującymi zmienną *wyjscie* w stan wysoki potwierdziły, że dla stanu wysokiego zmiennej *wyjscie* (kolor zielony koła, niebieska ramka – Rys. 6) nastąpiła zmiana na czwartym bajcie od końca pola danych (z 00 na 01). Tym samym znaleziono miejsce przesyłu wartości zmiennej w komunikacie.

Diagnozowanie komunikacji kontynuowano rozbudowując nieznacznie układ sterowania przez dodanie zmiennej *wyjscie*2, która przyjmować powinna wartość tożsamą ze zmienną *wyjscie* (Rys. 7). Celem powielenia zmiennych było zaobserwowanie, w jaki sposób przesyłane są dwie wartości zmiennych ze stacji procesowej do stacji operatorskiej.

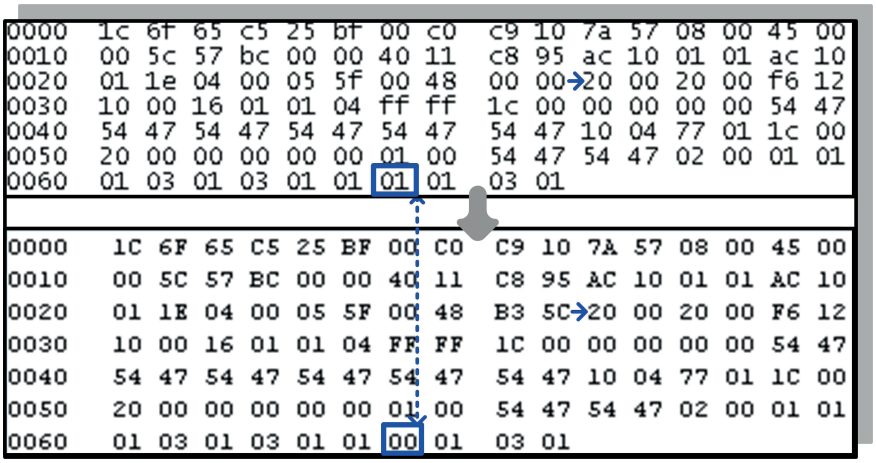
Na obrazie graficznym stacji operatorskiej dodano drugie koło. Kolor wyświetlania, analogicznie jak poprzednio, uzależniono od stanu zmiennej *wyjscie*2. W pierwszej fazie nie wysterowano wyjść układu i przesyłano stany niskie obydwu zmiennych. Na obrazie graficznym (Rys. 8) obydwa koła miały kolor czerwony, oznaczający transmisję stanu niskiego wartości zmiennych.

Zwrócono uwagę na kilka obserwowanych elementów:

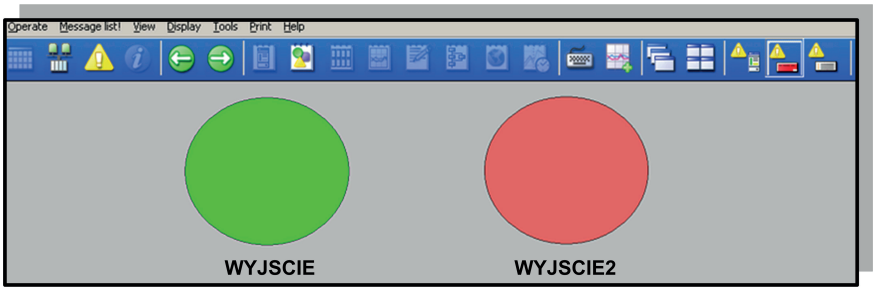
a) długość pola danych datagramu, która zmieniła się o cztery bajty (Rys. 8) w stosunku do przypadku przesyłu jednej wartości zmiennej;
b) czwarty bajt od końca i ósmy bajt od końca (wartości 00);
c) pierwszy (oznaczony strzałką) bajt pola danych i trzeci (wartość 20 w ramce).

Następnie przyporządkowano poziom wysoki sygnału (przyciskiem podłączonym do *start*) zmiennej *start*. Wynikiem wykonania układu z diagramu FBD (Rys. 7) było podanie na wyjścia *wyjscie* i *wyjscie*2 wartości logicznej „1” (TRUE).

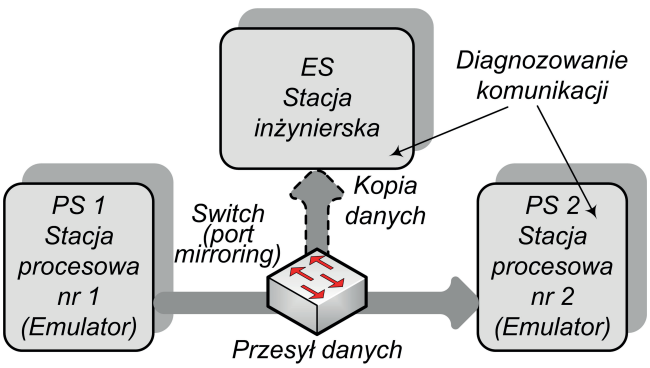
Potwierdzeniem przesłania wartości zmiennych wyjściowych była wizualizacja na odległej stacji operatorskiej zielonych kół (oznaczających *wyjscie* = *wyjscie*2 = 1 – Rys. 9). W wyniku transmisji dwóch wartości (*wyjscie* = *wyjscie*2 = 1) w przechwyconym polu danych na czwartym i ósmym bajcie od



Rys. 10. Zmienione dane w przypadku zewnętrznego zakłócenia przesyłu
Fig. 10. Changed data in the case of external transfer disruption



Rys. 11. Wizualizacja w przypadku zewnętrznego zakłócenia przesyłu
Fig. 11. Visualization in the case of external transfer disruption



Rys. 12. Diagnozowanie komunikacji pomiędzy stacjami procesowymi
Fig. 12. Diagnosing communication between process stations

końca zdiagnozowano wartości 01. Wykonując jeszcze kilkadziesiąt dodatkowych, innych niż opisane prób, zaobserwowano, że zmienne transmitowane są kolejno od czwartego bajtu od końca, co cztery bajty w stronę początku wiadomości (stąd potwierdzenie wyniku obserwacji „b)"). Zwiększa się też długość pola danych datagramu (z 60B na 64B – potwierdzenie wyniku obserwacji „c)")

3. Zakłócenie transmisji

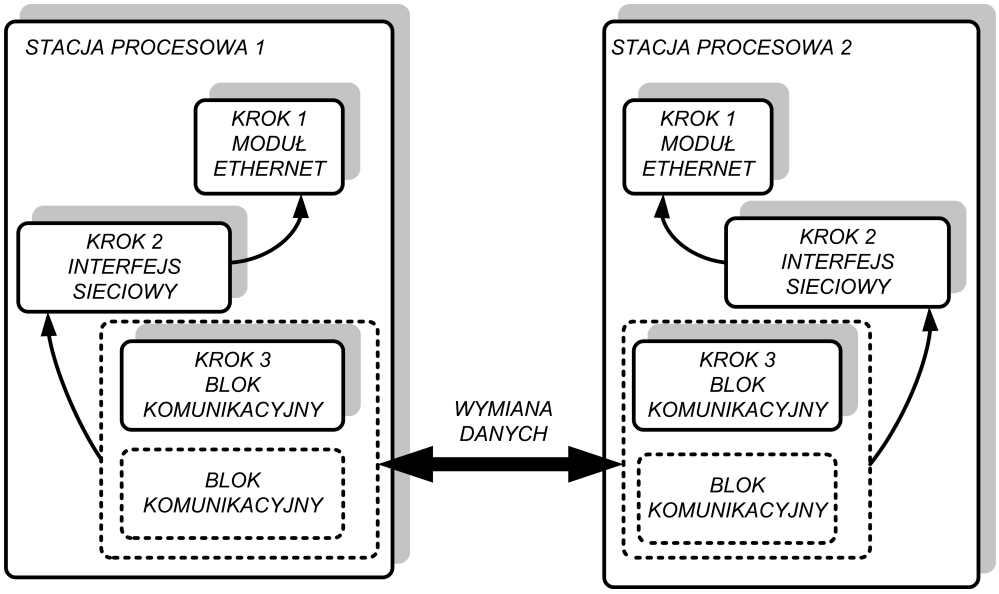
Stacja operatorska służy do wizualizacji działania procesu przebiegającego w stacji procesowej. Zatem operator powinien otrzymywać na ekranie synoptycznym wiarygodne dane. Należy zaznaczyć, że badany protokół komunikacyjny nie wyposażono w mechanizm obliczania sumy kontrolnej z wartości pola danych. Jedynym mechanizmem kontroli integralności jest zawarta w nagłówku TCP/UDP suma kontrolna liczona na podstawie całego komunikatu (a nie tylko z pola danych). Czy ewentualna podmiana wartości danych procesowych oraz umieszczenie danych w spreparowanym komunikacie będzie traktowane przez stację operatorską jako poprawne? Podczas prób zastanawiano się: co stałoby się, gdyby został zakłócony przesył danych? Gdyby pomimo wysterowania wyjść w stan wysoki, został zaburzony obraz? Czy jest to tu możliwe? Posiłkując się schematem FBD (Rys. 7) można z góry założyć, że pojawienie się różnych wartości zmiennych *wyjscie* i *wyjscie2* nie jest możliwe, są one bowiem dołączone do wspólnej linii sygnałowej.

W celu przetestowania możliwości zaburzenia wizualizacji, spreparowano odpowiedni datagram z podmienioną wartością czwartego bajtu od końca (zmiana z 01 na 00 – por. Rys. 10). Oczywiście zadbano, aby pojawiał się on około dwukrotnie częściej niż cykl nadawczy stacji procesowej i czas odświeżania. Stacja operatorska otrzymywała więc datagramy poprawne i – znacznie częściej – zaburzone, spreparowane (bezwładność odświeżania wizualizacji powodowała wyświetlanie stanu zgodnego z podmienionymi danymi). W wyniku przeprowadzonego eksperymentu, otrzymano obraz przedstawiony na Rys. 11. Zwizualizowano sytuację niemożliwą w przypadku zdatności systemu sterowania (jednoczesne różne wartości zmiennych na wyjściach), zatem skutecznie zakłócono transmisję!

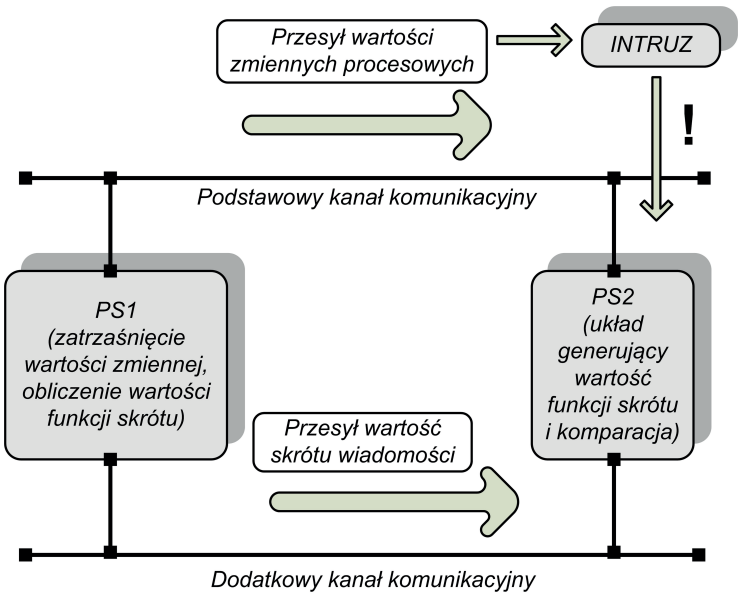
4. Diagnozowanie komunikacji PS-PS

W celu diagnozowania komunikacji pomiędzy stacjami procesowymi zestawiono układ przedstawiony na Rys. 12. W przypadku łączenia fizycznych stacji procesowych, diagnostykę przeprowadza się z poziomu stacji inżynierskiej. Jeśli w konfiguracji zamiast jednej ze stacji zostanie użyty emulator, to testowanie można przeprowadzić ze stanowiska emulatora. Zastosowano tu podobne „zabiegi" wstępne do przedstawionych w części 2.

W celu poprawnej komunikacji między stacjami procesowymi skonfigurowano układ komunikacji obydwu stron wykorzystujący bloki komunikacyjne FBD. Kolejne kroki procesu konfiguracji, obejmują: wybranie w strukturze sprzętowej modułu komunikacyjnego (Ethernet), przyporządkowanie odpowiedniego interfejsu sieciowego (określającego rodzaj protokołu UDP/TCP),



Rys. 13. Sposób konfiguracji komunikacji stacji procesowej
Fig. 13. Process station communication configuration method



Rys. 14. Propozycja dodatkowego zabezpieczenia przez użytkownika przesyłanych danych
Fig. 14. Proposed additional user-protection of transferred data

a następnie dodanie bloków komunikacyjnych wysyłających/odbierających dane odpowiednio z każdej strony (Rys. 13). Do testów wybrano ww. sposób zestawienia komunikacji: najpierw używając interfejsu UDP, potem – TCP. W wyniku przeprowadzonych doświadczeń i wielokrotnych przesyłów danych różnice nie były znaczące.

Proces diagnozowania komunikacji PS-PS przebiegał podobnie do fragmentu przedstawionego w punkcie drugim, dotyczącego komunikacji PS-OS. Ze względu na znaczną objętość wyników badań, najważniejsze wyniki wnioskowania dotyczącego ruchu sieciowego między stacjami procesowymi zebrano syntetycznie poniżej:

- stały pierwszy bajt pola danych (wartość 20);
- bajt rozmiaru przesyłanej zmiennej (nr 3) – pierwszy bajt rozmiaru danych;
- numeracja komunikatów – znaleziono także bajt przekazujący numer każdego kolejnego komunikatu (nr 13) – inkrementowany po każdym wysłaniu;
- bajt numeracji (nr 15) – zawiera ID modułu odbierającego;
- pierwszy bajt statusu (nr 17) – blok wysyłający ustawia na „00”;
- potwierdzenie odebrania – ustawienie w wiadomości zwrotnej bajtu statusu na „01”;
- bajt statusu czasu *timeout* (nr 19) – pierwszy bajt zawierający wartość czasu timeout;
- próba wysyłania wartości typu *integer* – 2-bajtowa wartość umieszczona jest na końcu pola danych (potwierdzenie jest krótsze o dwa bajty);
- format danych – wartości przesyłane są w formacie odwróconym (najpierw najmniej znaczące bajty).

5. Podsumowanie

W artykule przedstawiono wybrane fragmenty procesu diagnozowania komunikacji pomiędzy stacją procesową i operatorską, a także pomiędzy dwiema stacjami procesowymi mini-systemu rozproszonego. Na podstawie przedstawionego w punkcie drugim fragmentu badania można zauważyć, że proces diagnozowania jest bardzo pracochłonny. Badanie przeprowadzane jest właściwie na zasadzie „czarnej skrzynki". W wyniku przeprowadzonych eksperymentów uzyskano informacje dotyczące sposobu transmisji i położenia wartości zmiennych procesowych w przesyłanych komunikatach.

Informacje te można wykorzystać do skomunikowania stacji systemu z rozszerzającymi zasobami użytkownika. Przykładem może być brama (ang. *gateway*) PS – Modbus TCP, umożliwiająca komunikację między zewnętrznymi systemami i emulatorem, w którym nie zaimplementowano protokołu Modbus TCP.

Omówione w artykule wyniki diagnozowania komunikacji między stacją procesową i stacją operatorską pokazują wyraźnie, że przesył danych, wizualizujących proces sterowany przez stację procesową, jest realizowany bez dodatkowych zabezpieczeń. W celu zapewnienia większego bezpieczeństwa proponuje się zestawić połączenie stacja procesowa – emulator (Rys. 14).

Na stanowisku emulatora stacji procesowej powinno być uruchomione oprogramowanie wizualizacyjne wymieniające dane lokalnie w ramach jednego fizycznego stanowiska. Oprócz danych zawierających wartości zmiennych procesowych, alternatywnym kanałem komunikacyjnym (fizycznym lub logicznym) może być przesłana wartość funkcji skrótu pierwotnej wiadomości, obliczona z użyciem uzgodnionego parametru ziarna. Układ komparacyjny w stacji odbiorczej po wykonaniu analogicznych obliczeń, porównywałby przesłane i obliczone wartości funkcji skrótu. Tylko w przypadku zgodności, wartość zmiennej procesowej byłaby wizualizowana na stacji operatorskiej. Nie zapobiegnie to oczywiście celowym, wyrafinowanym atakom na integralność przesyłanych komunikatów, ale zminimalizuje możliwość wizualizacji niewłaściwych danych.

Bibliografia

1. Bednarek M., Dąbrowski T., Olchowik W., *Selected practical aspect of communication diagnosis in the industrial network,* „Journal of KONBiN", Nr 49, 2019, 383–404, DOI 10.2478/jok-2019-0020.
2. Bednarek M., Dąbrowski T., *Trójpoziomowe zabezpieczanie integralności i poufności przesyłanych danych w sieci przemysłowej*, „Biuletyn WAT", Vol. LXVI, nr 1, 2017, 81–90, DOI: 10.5604/01.3001.0009.9486.
3. Bednarek M., Będkowski L., Dąbrowski T., *Komparacyjno-progowe diagnozowanie w systemie transmisji komunikatów*, „Przegląd Elektrotechniczny", nr 5, 2008, 320–324.
4. Bacidore M., *Ethernet vs. fieldbus: the right network for the right application*, www.controldesign.com, 17.10.2016.
5. Chipkin P., *Modbus for Field Technicians*, Createspace Independent Publishing Platform, 2011.
6. Goldstein A., *Tutorial on Modbus RTU Slave Design: How to Design Modbus Slave Device in C*, Amazon Digital Services, US, 2020.
7. Solnik W., Zajda Z., *Sieć Profibus DP w praktyce przemysłowej. Przykłady zastosowań,* Wydawnictwo BTC – Korporacja, e-book, Legionowo, 2013.
8. Bender K., *PROFIBUS: Fieldbus for Industrial Automation*, Prentice-Hall, 1993.
9. Danielis P., Skodzik J., Altmann V., Schweissguth E.B., Golatowski F., Timmermann D., Schach J., *Survey on real-time communication via ethernet in industrial automation environments*, Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), 2014, DOI: 10.1109/ETFA.2014.7005074.
10. Spurgeon Ch.E., Zimmerman J., *Ethernet: The Definitive Guide: Designing and Managing Local Area Networks*, O'Really Media, 2014.
11. Ansari S., Rajeev S.G., Chandrashekar H.S., *Packet sniffing: a brief introduction*, „IEEE Potentials", Vol. 21, No. 5, 2003, 17–19, DOI: 10.1109/MP.2002.1166620.
12. Leslie F., Sikos L.F., *Packet analysis for network forensics: A comprehensive survey*, „Forensic Science International: Digital Investigation", Vol. 32, 2020, DOI: 10.1016/j.fsidi.2019.200892.
13. Ackerman P., *Industrial Cybersecurity*, Packt Publishing, Birmingham-Mumbai, 2021.
14. Coutu M., *The Technicians Guide to Modbus TCP*, Amazon Digital Services LLC – KDP Print US, 2020.
15. Popp M., *Industrial communication with Profinet*, e-book, Profibus & Profinet International (PI), 2018.
16. Stevens R., *TCP/IP Illustrated. The Protocols*, Vol. 1, Addison-Wesley 2012.
17. Syed M., *Black box thinking*, John Murray Press, 2016.
18. Cheswick W.R., Bellovin S.M., Rubin A.D., *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 2003.
19. Dawson R., Boyd C., Dawson E., Nieto J.M.G., *Skma: a key management architecture for scada systems*, „ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research", Australian Computer Society, Inc., 183–192, Darlinghurst, Australia, 2006.
20. Sorensen J., T., *Security in Industrial Networks*, Norwegian University of Science and Technology, July 2007.
21. Dąbrowski T., *Diagnozowanie systemów antropotechnicznych w ujęciu potencjałowo-efektowym*, Wydawnictwo Wojskowej Akademii Technicznej, Warszawa, 2001.
22. Bednarek M., *Wizualizacja procesów – laboratorium*, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2004.

Inne źródła

23. Tob, *Profinet – podstawy, teoria i praktyka*, https://iautomatyka.pl.
24. Profibus & Profinet International (PI), *PROFINET Technology and Application – System Description*, e-book, 2018.
25. Dokumentacja techniczna *Freelance. Getting Started*, Version 9.2 SP1, ACC, 2010.
26. *System 800xA. AC800M Communication Protocols*, ABB, 2003–2016.
27. Byres E.J., Hoffman D., Kube N., *On shaky ground – a study of security vulnerabilities in control protocols. Tech. rep.*, Wurldtech Analytics Inc, Wurldtech Analytics Inc. 208-1040 Hamilton St. Vancouver BC Canada V6B 2R9, 2006.
28. Bryes E., Karsch J., Carter J., *Firewall deployment for Scada and process control networks. Tech. Rep. 1.4*, NISCC, National Infrastructure Security Co-ordination Center P O Box 832 London, February 2005.
29. Beaver C.L., Gallup D.R., NeuMann W.D., Torgerson M.D., *Key management for Scada. Tech. Rep. SAND2001-3252*, Sandia National Laboratories, Sandia National Laboratories Albuquerque NM 87185-0785, Mai 2003.
30. Norma IEC 61131-3:2013. *Programmable controllers – Part 3: Programming languages.*

Diagnosis of Communication Between the Elements of a Distributed Control System

Abstract: The article describes selected fragments of the process diagnosing communication between a process and operator station, and between process stations of a distributed mini-system based on an AC800F modular industrial controller. Conducted experiments provided information on the transmission method and location of process variable values in transferred messages. The information can be used to make a decision regarding additional transfer protections or communicating system stations with expanding user resources.

Keywords: industrial networks, distributed control system, process station, operator station, communication diagnostics, data transfer

dr inż. Marcin Bednarek
bednarek@prz.edu.pl
ORCID: 0000-0001-8987-5134

Pracuje na stanowisku adiunkta w Katedrze Informatyki i Automatyki Wydziału Elektrotechniki i Informatyki Politechniki Rzeszowskiej. Stopień doktora nauk technicznych uzyskał w Wojskowej Akademii Technicznej. Główny obszar zainteresowań oraz działalności naukowej i dydaktycznej to: diagnostyka i eksploatacja systemów antropotechnicznych, komunikacja w rozproszonych systemach sterowania i sieciach przemysłowych, niezawodność i bezpieczeństwo systemów i sieci komputerowych. Jest autorem/współautorem ponad 130 publikacji.