Diagnozowanie komunikacji między stacjami procesowymi 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 stacjami procesowymi mini systemu rozproszonego, zbudowanego na bazie modułowego sterownika przemysłowego AC 800F. W wyniku przeprowadzonych eksperymentów uzyskano dane dotyczące sposobu transmisji i położenia wartości zmiennych procesowych w przesyłanych komunikatach. Informacje te można wykorzystać w przyszłości do skomunikowania stacji systemu z bramą rozszerzającą możliwości komunikacyjne.

Słowa kluczowe: diagnoza, komunikacja, przesył danych, rozproszony system sterowania, sieci przemysłowe, stacja procesowa

1. Wprowadzenie

Rozważania przedstawione w artykule dotyczą komunikacji między stacjami rozproszonego systemu sterowania DCS (ang. *Distributed Control System*) [1]. Stacjami systemu DCS mogą być w szczególności: stacje procesowe PS (ang. *Process Station*), stacje inżynierskie ES (ang. *Engineering Station*), stacje operatorskie OS (ang. *Operator Station*), stacje diagnostyczne DS (ang. *Diagnostic Station*) oraz stacje-bramy GS (ang. *Gateway Station*) [2, 3]. Opis funkcji poszczególnych stacji zebrano syntetycznie w Tabeli 1.

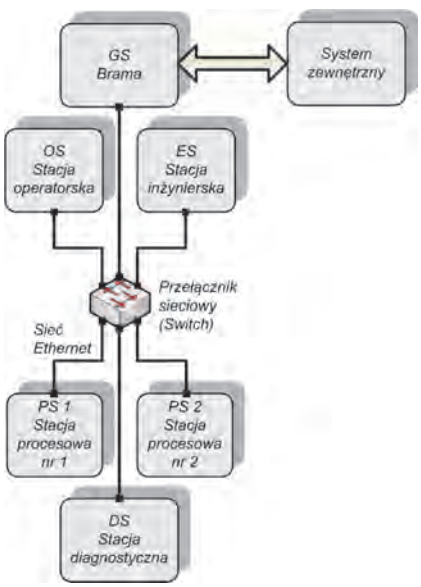
Stacje procesowe są sterownikami przemysłowymi realizującymi programy sterowania. Odpowiednie przedstawienie operatorowi przebiegu procesu w postaci statycznej grafiki z elementami dynamicznymi (obszary wypełniane, wyświetlacze alfanumeryczne, listy alarmów i podpowiedzi) oraz umożliwienie mu oddziaływania spoczywa na stacjach operatorskich. Z poziomu stacji inżynierskich uruchamia się i prowadzi proste testowanie poprawności działania stacji. Stacje diagnostyczne umożliwiają diagnozowanie działania stacji oraz komunikacji między nimi. Z punktu widzenia otwartości systemu [4] ważną rolę pełnią stacje-bramy GS. Dzięki nim możliwa jest wymiana danych z innymi systemami. Spełniają one rolę połączenia z zewnętrznym systemem, wykorzystującym standard komunikacyjny, np. brama OPC [5] w systemie Freelance ABB [16], a niezaimplementowanym w systemie źródłowym. Stacje systemu mogą być połączone magistralą komunikacyjną lub sieć komunikacyjna może tworzyć strukturę gwiaździstą. W ostatnim przypadku (Rys. 1) najbardziej popu-

Tabela 1. Funkcje stacji DCS
Table 1. DCS station functions

Stacja	Skrót	Funkcja
procesowa	PS	Sterowanie procesem przemysłowym
operatorska	OS	Wizualizacja procesu, oddziaływanie operatorskie, archiwizacja, alarmowanie
inżynierska	ES	Konfiguracja systemu, uruchamianie stacji
diagnostyczna	DS	Diagnozowanie stacji systemu, diagnozowanie komunikacji
brama	GS	Połączenie z innymi systemami (protokołami)

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

Artykuł recenzowany
nadesłany 02.01.2024 r., przyjęty do druku 20.02.2024 r.



Rys. 1. Rozpatrywany system wykorzystujący standard Ethernet
Fig. 1. System under consideration using the Ethernet standard

larnym rozwiązaniem jest wykorzystanie sieci bazującej na standardzie Ethernet [6] za pomocą przełączników (ang. *switch*) [7].

Część pierwsza opracowania [8] poświęcona była diagnozowaniu komunikacji między stacją procesową i operatorską systemu oprogramowania sterownika AC 800F [9]. Diagnozowanie prowadzone było pod kątem znalezienia odpowiedzi na pytanie: „*czy zamknięty i nieudokumentowany protokół komunikacyjny może dawać poczucie bezpieczeństwa – nienaruszalności i poufności transmisji?*". W wyniku przeprowadzonych eksperymentów i przedstawionych wyników potwierdziła się druga zasada Kerckhoffsa [10, 11], gdzie tajność i zamknięty charakter protokołu wymiany danych daje jedynie złudne poczucie bezpieczeństwa.

Przewodnią myślą niniejszej pracy jest próba odpowiedzi na pytanie: „*w jaki sposób stacje procesowe należące do rozdzielnych systemów klasy DCS wymieniają wzajemnie dane*?" i czy, tym samym, pozyskując dane dotyczące sposobu komunikacji użytkownik może zaprojektować stację-bramę, która mogłaby być ogniwem spajającym z systemami zewnętrznymi? Nie jesteśmy teraz zatem zainteresowani aspektami bezpieczeństwa, a próbujemy odgadnąć sposób komunikacji stacji procesowych. Problem ten, podobnie jak poprzednio [8], sprowadza się do zagadnienia badania „czarnej skrzynki" [12]. W tym przypadku, oprócz danych wejściowych i wyjściowych (komunikaty), mamy możliwość zmiany także niektórych parametrów konfiguracyjnych z poziomu stacji nadającej.

2. Komunikacja między stacjami procesowymi

W rozpatrywanym systemie istnieje kilka rodzajów wzajemnej komunikacji stacji procesowych [16]. Wymienione tu zostaną dwa najważniejsze.

Najprostszym sposobem jest użycie zmiennych wewnętrznych projektu. W przypadku, gdy wszystkie stacje procesowe są zasobami wspólnego DCS, można zastosować pierwszy ze sposobów komunikacji. Wystarczy zaznaczenie, podczas tworzenia zmiennej, opcji „export", aby widoczna była ona dla pozostałych PS. Tego typu metoda komunikacji nie nadaje się do zastosowania do ewentualnej wymiany danych z zewnętrznymi systemami i bramami wykonanymi przez użytkownika. Budowa komunikatów jest wprawdzie poznana przez autora (lecz nieudokumentowana firmowo) i określenie położenia danych w komunikatach jest możliwe, jednak ze względu na zmieniającą się liczbę zmiennych w zależności od wersji skompilowanego pliku projektu eksportowanego do stacji systemu, ma zmienną długość i zmienne położenia pól danych [8]. Sposób ten nie nadaje się do wykorzystania.

Drugim ze sposobów skomunikowania PS-PS jest użycie specjalnie przygotowanych bloków komunikacyjnych. Metoda ta wymaga zdefiniowania w drzewie projektu, przedstawiającego strukturę DCS, kolejno kilku elementów-zasobów (Rys. 2):
- w strukturze sprzętowej każdej ze stacji (przynależnej do dwóch różnych systemów) umieszcza się (definiuje) *moduł komunikacji* Ethernet;
- następnie do modułu przyporządkowuje się (definiuje, wybierając dostępny element z menu) odpowiedni *interfejs komunikacyjny*;
- dla każdego z interfejsów komunikacyjnych przypisuje się w projekcie odpowiedni *blok nadawczy* lub *odbiorczy*, w zależności po której stronie następuje konfiguracja (tutaj: PS1 – blok nadawczy, PS2 – blok odbiorczy).

Na rysunku 2 przedstawiono graficznie ww. opisane kroki. Tak zdefiniowanych par nadawczo-odbiorczych bloków komunikacyjnych może być kilka.

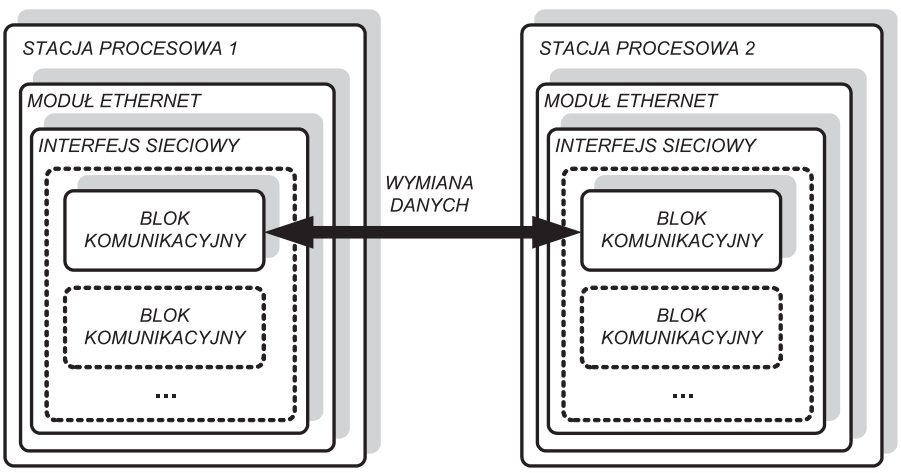
Wśród możliwych do użycia interfejsów komunikacyjnych są:
- *SR_SRTCP* – stosowany do połączenia za pomocą protokołu TCP, który umożliwia komunikację tzw. połączeniową, tworząc wirtualny kanał komunikacyjny [13];
- *SR_SNDEV* – interfejs wysyłający stosowany do ustanowienia połączenia za pomocą protokołu UDP, znacznie prostszego w implementacji [13];
- *SR_RNDEV* – interfejs odbierający za pomocą protokołu UDP.

Więcej na temat sposobów komunikacji prowadzonej z zastosowaniem rodziny protokołów TCP można znaleźć w [13].

Jak wynika z rysunku 2, ostatnim (najniższym w hierarchii) elementem potrzebnym do skonfigurowania komunikacji między stacjami procesowymi tą metodą jest blok komunikacyjny. Jest to element biblioteki predefiniowanych bloków komunikacyjnych ze względu na używany do konfiguracji stacji procesowej język FBD (ang. *Function Block Diagram*) [17]. Do standardowej komunikacji potrzebne są bloki:
- *SR_USEND* – blok nadawczy;
- *SR_ URECV* – blok odbiorczy.

Do ustanowienia komunikacji PS-PS niezbędna jest konfiguracja pary nadawczo-odbiorczej.

Właśnie drugi z opisanych sposobów jest interesujący z punktu widzenia skomunikowania stacji procesowej z systemami zewnętrznymi. Należy w tym celu wykonać „małą podmianę". Zamiast odbiorczej stacji procesowej skonfigurować własną stację-bramę zaprogramowaną przez użytkownika jako miejsce połączenia z zewnętrznym systemem. Tylko pozornie chodzi tu o małą zmianę. Przecież do skomunikowania się

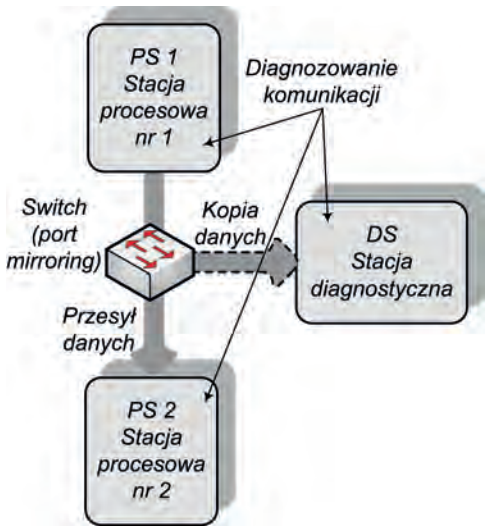


Rys. 2. Hierarchiczny sposób konfiguracji komunikacji stacji procesowych
Fig. 2. Process stations communication configuration hierarchical method

z zewnętrznym systemem należy tak zaprojektować i zbudować oprogramowanie stacji-bramy, aby nadająca stacja procesowa nie odczuła podmiany i traktowała drugą stronę jak zwykłego partnera komunikacyjnego. Naturalnie w tym celu potrzebne jest zdiagnozowanie komunikacji PS-PS prowadzonej drugim z wymienionych sposobów. Celem diagnozowania jest poznanie zawartości i przeznaczenia przynajmniej najważniejszych pól komunikatów przesyłanych między stacjami.

3. Diagnozowanie komunikacji PS-PS

Do zbadania komunikatów przesyłanych między stacjami należy zbudować układ diagnostyczny (podobny do przedstawionego w [8]). W tym celu należy użyć dwóch komunikujących się stacji procesowych PS1 i PS2 (Rys. 3) oraz stacji diagnostycznej, której oprogramowanie *sniffera* przechwytywałoby przesyłane wiadomości [14]. W tak skonfigurowanym układzie diagnostycznym (Rys. 3) konieczne byłyby dodatkowe zabiegi polegające na utworzenia w urządzeniu łączącym (przełączniku) tzw. portu lustrzanego [14], na który wysyłany byłby cały ruch sieciowy. Spowodowane to jest specyficznym działaniem przełącznika (*switch*), który ogranicza domeny kolizyjne do dwóch komunikujących się urządzeń i w standardowej konfiguracji nie daje możliwości podsłuchu z zewnątrz.

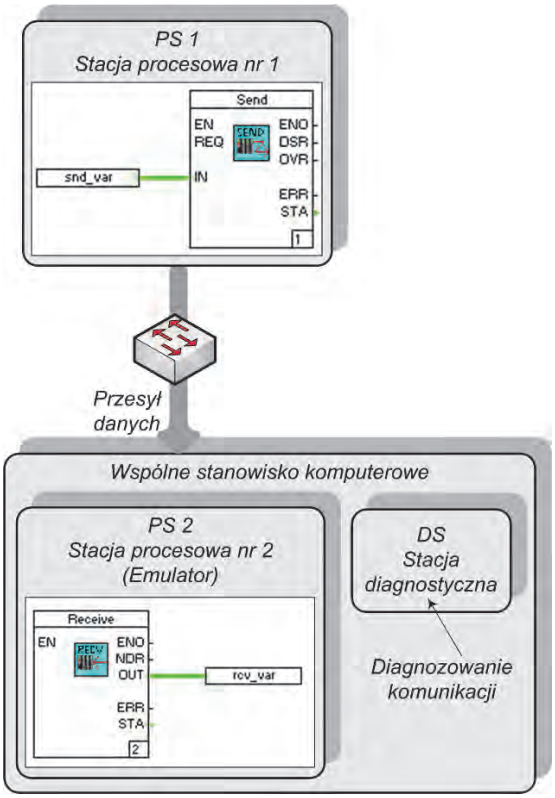


Rys. 3. Diagnozowanie komunikacji między stacjami procesowymi
Fig. 3. Diagnosing communication between process stations

W celu uproszczenia układu diagnostycznego do badania komunikacji użyto zmodyfikowanego wspólnego stanowiska komputerowego (Rys. 4). Ograniczono tu liczbę stacji do dwu. Zamiast stacji PS2 wykorzystano emulator działający w środowisku Windows komputera. Tam też zainstalowano oprogramowanie *sniffera* stacji diagnostycznej, przechwytujące komunikaty z lokalnego portu komunikacyjnego Ethernet. Dzięki temu nie było konieczności przekierowywania ruchu sieciowego w przełączniku i innych dodatkowych działań.

W stacji PS1 skonfigurowano możliwie najprostszy schemat komunikacji FBD (Rys. 4), zawierający etykietę zmiennej wejściowej (wysyłanej) połączonej do bloku nadawczego, w stacji PS2 – analogiczny układ odbiorczy: blok odbiorczy i połączona etykieta zmiennej wyjściowej (odbieranej). Ustawiono także wstępne, zmieniane później wielokrotnie w trakcie diagnozowania, parametry komunikacji (*timeout* = 1136 ms oraz ID = 8 modułu odbierającego, *snd_var* = 10).

W jaki sposób diagnozowano komunikację PS-PS? W tak skonfigurowanym środowisku diagnostycznym, wykonano szereg prób przesyłu danych między PS1 i emulatorem PS2 z użyciem



Rys. 4. Diagnozowanie komunikacji między stacją procesową i emulatorem PS
Fig. 4. Diagnosing communication between process stations and PS emulator

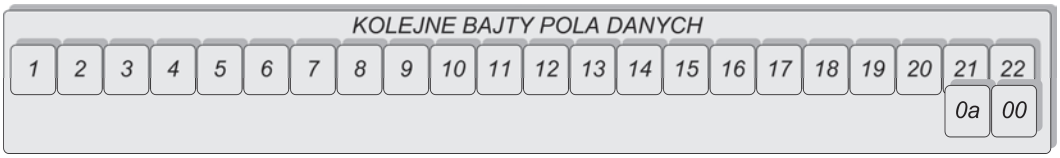
bloków nadawczych i odbiorczych. Proces diagnozowania opierał się na wielokrotnym powtarzaniu schematu:

- ustawienia wartości pewnego parametru konfiguracyjnego (np. przesyłanego w wiadomości);
- dokonania przesyłu testowego i zarejestrowania komunikatu;
- zmiany wartości parametru;
- dokonania powtórnego przesyłu testowego i zarejestrowania komunikatu;
- porównania zapisanych komunikatów – obserwacji zmiany wartości poszczególnych pól komunikatu;
- zwrócenie uwagi na miejsce zmiany przesyłanej początkowej i zmienionej wartości.

Dzięki wykonaniu kilkudziesięciu prób zmian każdego z możliwych parametrów komunikacyjnych, a w każdej próbie kilkuset tysięcy cyklicznych przesyłów, oprócz znalezienia znaczenia pól komunikatu, zaobserwowano pewne stałe i niezmienne wartości pola danych komunikatu. Zmiana ustawień konfiguracji zarówno układu FBD, jak i modułu komunikacyjnego, czy też bloków nadawczego i odbiorczego nie powodowała różnic w kilku miejscach pola danych. Były to bajty o numerach: 2, 5–12, a także 16 i 18. Wymienione pojedyncze bajty miały wartość zero („00”), natomiast sekwencja – stałą strukturę o wartości różnej od zera. Przykładowe wyniki diagnozowania komunikacji podano poniżej.

3.1. Próba 1. Długość pola danych i położenie wartości zmiennej typu całkowitego

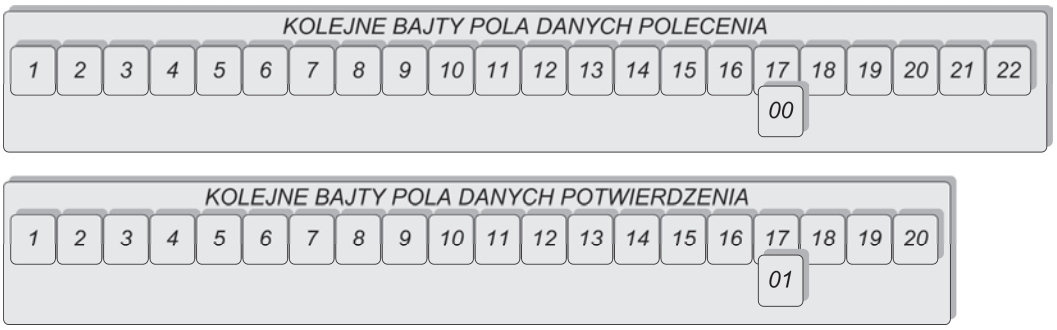
Podłączono zmienną dwubajtową typu całkowitego. Zestawiono połączenie wysyłając kolejno wartości 0 i 10. Zaobserwowano komunikat o długości 22 bajtów – zaobserwowano zmiany (0/a Hex) na 21 bajcie. Czynność powtórzono dla kilku innych wartości. W ten sposób udało się znaleźć położenie wartości zmiennej (na 21 i 22 bajcie w formacie odwróconym). Na Rysunku 5 przedstawiono zdiagnozowane położenie wartości zmiennej w polu danych.



Rys. 5. Ilustracja próby pierwszej
Fig. 5. Illustration of the first attempt

3.2. Próba 2. Potwierdzenie otrzymania

Podczas wielokrotnych prób wysyłania wartości zmiennej zauważono pierwszą ważną zasadę, na której bazuje wymiana danych – istnienie mechanizmu potwierdzania dostarczenia wartości zmiennej. Na Rysunku 6 pokazano zmianę wartości 0/1 siedemnastego bajtu pola danych. Odpowiedź PS2 jest krótsza (20B) i nie zawiera przesyłanej wartości zmiennej.



Rys. 6. Ilustracja próby drugiej
Fig. 6. Illustration of the second attempt

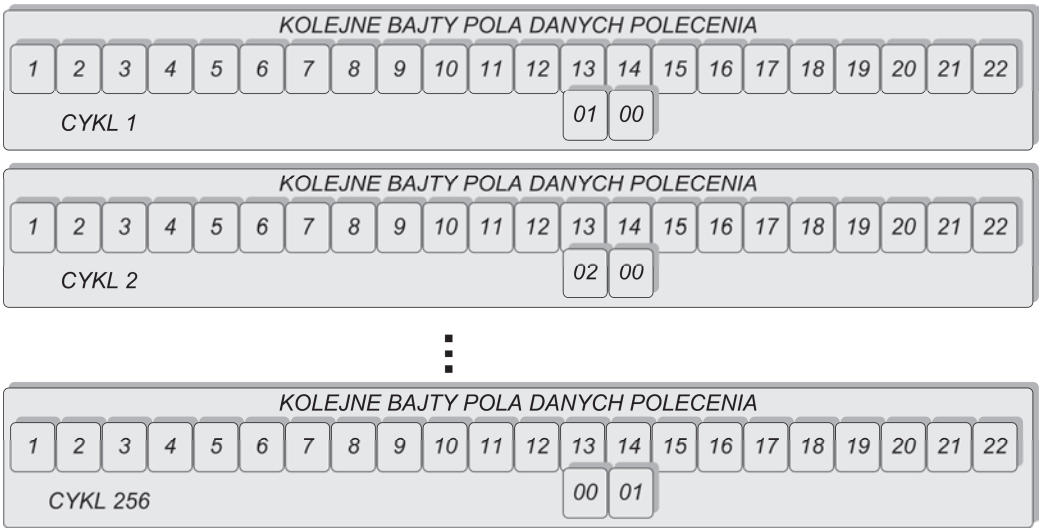
3.3. Próba 3. Oznaczenie typu przesyłanej wartości i długości pola komunikatu zawierającego przesyłaną wartość

Podczas kolejnych wielu prób wysyłania wartości zmieniano typ (długość) przesyłanej zmiennej. Na podstawie wnioskowania diagnostycznego zaobserwowano zmiany na trzecim bajcie pola danych komunikatu. Wykonano to dla zmiennych przechowujących wartości binarne oraz zajmujących cztery bajty (w tej implementacji systemu DWORD ma 4B) W przypadku badania przesyłu większych struktur, wcześniej zdefiniowano struktury o długości od 128B do 256B. Rysunek 7 ilustruje powyższe rozważania. Symbolem klamry zaznaczono miejsca położenia przesyłanych wartości w zależności od liczby bajtów zajmowanych w komunikacie. Ogólnym wnioskiem jest znalezione położenie wartości zmiennej na końcu pola danych. Na trzecim bajcie zaobserwowano transmisję wartości reprezentującej długość pola zawierającego przesyłaną wartość. W przypadku większych struktur, wartość długości danej umieszcza się także na czwartym bajcie pola danych. Po przeprowadzonej próbie trzeciej należy zmodyfikować wniosek z punktu 3.1: długość pola danych polecenia jest uzależniona od typu przesyłanej zmiennej przechowującej przesyłaną wartość (20B powiększone o wartość reprezentującą długość danej).

3.4. Próba 4. Kolejność wysyłanych danych

Ze względu na stosowaną w systemach automatyki i dozorowania cykliczność wysyłania, sprawdzono komunikaty pod kątem numeracji kolejnych przesyłanych wartości danej zmiennej procesowej. Komunikaty muszą być numerowane, ponieważ stacje pracując w rozległej sieci przemysłowej, w której występują *routery*, mogą mieć różne trasy przesyłu komunikatów. Wiąże się to z możliwością nadejścia komunikatu później wysłanego przed wcześniejszym. Aby zabezpieczyć system przesyłu przed przypadkowym nadpisaniem aktualnej wartości zmiennej procesowej wartością nieaktualną (poprzednią) kolejne cykle wysyłania wartości są numerowane (bajt 13 i 14 na Rys. 8).

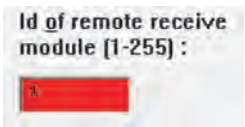
Rys. 7. Ilustracja próby trzeciej
Fig. 7. Illustration of the third attempt



Rys. 8. Ilustracja próby czwartej
Fig. 8. Illustration of the fourth attempt

3.5. Próba 5. Identyfikator modułu odbierającego

Jak wspomniano w sekcji 2. artykułu, w systemie może funkcjonować wiele par „blok nadawczy–blok odbiorczy". Maski konfiguracyjne bloków mają parametr obowiązkowy, którym jest ID modułu odbiorczego (Rys. 9 – fragment okna konfiguracyjnego modułu nadawczego).



Rys. 9. Fragment maski konfiguracyjnej bloku nadawczego – ID odbiornika
Fig. 9. Fragment of the configuration mask of the transmitter block – receiver ID

Zmieniając kilkukrotnie w bloku nadawczym parametr ID odbiornika (por. Rys. 10 – zmiana ID z wartości 8 na 15), zaobserwowano następną zasadę wymiany danych: zmiany piętnastego bajtu pola danych (Rys. 10). Wartość ID nie może być większa niż 255, dlatego na tej obserwacji zaprzestano badania wpływu zmiany ID na zawartość komunikatu.

W odpowiedzi zwrotnej można zaobserwować z kolei stałe ustawienie bajtu nr 15 na wartość 0×fe. Kolejne próby potwierdziły brak zmian w odpowiedzi PS2. Jest to kolejna flaga (po bajcie nr 17) ustawiana przez blok odbierający przesyłana w komunikacie zwrotnym.

3.6. Próba 6. Parametr timeout

W bloku odbiorczym można ustawić czas przerwania połączenia w przypadku nieotrzymania komunikatu (*timeout*). Zmieniając jego wartość, wielokrotnie zaobserwowano, że moduł nadawczy przesyła go modułowi odbiorczemu na bajtach numer 19 i 20 (także jak wszystkie wcześniejsze wartości w zapisie-formacie odwróconym, tj. najpierw mniej znaczący bajt, a później bajt bardziej znaczący – por. przykład na Rys. 11). Jest to kolejna ważna informacja dotycząca sposobu komunikacji.

Rys. 10. Ilustracja próby piątej
Fig. 10. Illustration of the fifth attempt

Rys. 11. Ilustracja próby szóstej
Fig. 11. Illustration of the sixth attempt

4. Podsumowanie wyników diagnozowania

Diagnozowanie komunikacji między stacjami procesowymi jest doświadczeniem bardzo czasochłonnym i wymagającym. W wyniku przeprowadzonych badań zdiagnozowano kilka kluczowych zasad, rządzących procesem wymiany danych podczas komunikacji PS-PS. Należą do nich dodatkowe, niezależne od stosowanego protokołu, mechanizmy potwierdzania (p. 3.2 i 3.5) i numeracji kolejnych komunikatów (p. 3.4), zastosowanie przesyłu parametru *timeout* (p. 3.6) i zwrotnych flag informacyjnych oraz znaczenie kolejnych bajtów pola danych przesyłanego komunikatu.

Podsumowując, pole danych komunikatu PS-PS, wykorzystującego do przesyłu pary bloków nadawczo-odbiorczych, ma następującą strukturę:

bajty 1 i 2	–	wartość stała (wartość 20 = 0×14) oznaczająca długość pola danych sterujących oraz bajt uzupełnienia („00”);
bajty 3 i 4	–	wielkość przesyłanej zmiennej; długość całego pola danych to suma wartości zapisanych na 1. oraz 3. i 4. bajcie;
bajty 5–12	–	bajty zarezerwowane (znana, niezmienna sekwencja ośmiu bajtów);
bajty 13 i 14	–	numeracja komunikatów; wartość inkrementowana w każdym kolejnym cyklu nadawczym;
bajty 15 i 16	–	ID modułu odbierającego lub stała wartość (0×fe) ustawiana w komunikacie zwrotnym oraz bajt uzupełnienia („00”);
bajty 17 i 18	–	bajt statusu – blok wysyłający ustawia na „00”; blok odbierający ustawia potwierdzenie odebrania w wiadomości zwrotnej na „01” oraz bajt uzupełnienia („00”);
bajty 19 i 20	–	wartość parametru *timeout*;
bajt 21 i kolejne	–	przesyłana wartość o długości określonej w bajtach 3 i 4;

Należy pamiętać, że komunikat potwierdzający ma długość 20B oraz zwrócić uwagę na format danych – wartości przesyłane są w formacie odwróconym (najpierw najmniej znaczące bajty). Zdiagnozowanie znaczenia kolejnych bajtów pola danych komunikatu pozwoliło na połączenie systemu AC 800F z bramą konwertującą komunikaty na inny standard przemysłowy (Modbus TCP [15]). Wykorzystanie emulatora stacji procesowej, bloków funkcjonalnych definiowanych przez użytkownika [18] oraz specjalnej dedykowanej bramy odbierającej komunikaty z bloku nadawczego, umożliwiło wykonanie standardowo nieobsługiwanego połączenia z zewnętrznym systemem.

Bibliografia

1. Stój J., *Wybrane zagadnienia sieci komunikacyjnych w przemysłowych systemach komputerowych*, Wydawnictwo Politechniki Śląskiej, Gliwice 2023.
2. Bednarek M., Dąbrowski T., Olchowik W., *Selected practical aspect of communication diagnosis in the industrial network,* „Journal of KONBiN”, Vol. 49, No. 1, 2019, 383–404, DOI: 10.2478/jok-2019-0020.
3. 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.
4. Klonowski Z.J., *Systemy informatyczne zarządzania przedsiębiorstwem. Modele rozwoju i właściwości funkcjonalne*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2004.
5. Mahnke W., Leitner S.-H., Damm M., *OPC Unified Architecture*, Springer-Verlag Gmbh, Berlin 2009.
6. Spurgeon Ch.E., Zimmerman J., *Ethernet: The Definitive Guide: Designing and Managing Local Area Networks*, O'Really Media, 2014.
7. 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
8. Bednarek M., *Diagnozowanie komunikacji między elementami rozproszonego systemu sterowania*, „Pomiary Automatyka Robotyka”, R. 26, Nr 4, 2022, 91-98, DOI: 1014313/PAR_246/91.
9. Bednarek M., *Wizualizacja procesów – laboratorium*, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2004.
10. Kerckhoffs A., *La cryptographie militaire,* „Journal des sciences militaires”, Vol. IX, 1883, 5–38 II, Desiderata de la cryptographie militaire.
11. van Tilborg H.C.A., Jajodia S. (eds.), *Encyclopedia of Cryptography and Security*, Springer 2011.
12. Syed M., *Black box thinking*, John Murray Press, 2016.
13. Fall K.R., Stevens R.W., *TCP/IP Illustrated Volume 1: The Protocols*, Addison Wesley, 2012.
14. 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.
15. Coutu M., *The Technicians Guide to Modbus TCP*, Amazon Digital Services LLC – KDP Print US, 2020.

Inne źródła

16. *System 800xA. AC 800M Communication Protocols*, ABB, 2003-2016.
17. Norma IEC 61131-3:2013. *Programmable controllers – Part 3: Programming languages.*
18. Dokumentacja techniczna *Freelance. Getting Started*, Version 9.2 SP1, ACC, 2010.

Diagnosing of Communication Between Process Stations of a Distributed Control System

Abstract: The article describes selected fragments of the process diagnosing communication between process stations of a mini-DCS based on an AC 800F 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 communicate the system's stations with a gateway to expand communication capabilities in the future.

Keywords: communication, diagnostics, data transfer, DCS, industrial network, process station

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.