Sieć, PLC, zintegrowane środowisko programistyczne, SRK, bezpieczeństwo

Jerzy MIKULSKI[1]
Szymon SURMA[1]

BEZPIECZEŃSTWO WARSTWY SIECIOWEJ ZINTEGROWANYCH SYSTEMÓW STEROWANIA

Artykuł opisuje zagadnienia z zakresu bezpieczeństwa warstwy sieciowej systemów sterowania opartych na sterownikach PLC. Porusza również kwestie różnic w systemach scentralizowanych i rozproszonych, głównie pod kątem organizacji wymiany informacji pomiędzy elementami systemu. Analizie poddano również zasadność wielowarstwowej integracji oprogramowania przeznaczonego do budowy systemów sterowania. Powyższe zagadnienia zastosowano do systemów sterowania ruchem kolejowym.

NETWORK LAYER SAFETY OF INTEGRATED CONTROL SYSTEMS

The article describes the issues of network layer safety of control systems based on PLC. It also raises issues of differences in centralized and distributed systems in scope of the organization for the information exchange between system elements. The merits of multi-integration software designed for building control systems was analyzed. Above mentioned issues were applied to the railway traffic control systems.

1. WSTĘP

Bezpieczeństwo w systemach sterowania może być rozumiane w różny sposób. Jedną z interpretacji pojęcia bezpieczeństwo jest bezpieczeństwo systemu. Bezpieczeństwo systemu rozumiane jest, jako brak występowania nieakceptowanych poziomów ryzyka szkód w czasie. Zgodnie z budową systemów sterowania, na bezpieczeństwo systemu składa się bezpieczeństwo sprzętowe i bezpieczeństwo programowe.

2. BEZPIECZEŃSTWO

Oprogramowanie składa się z dwóch podstawowych warstw - systemu operacyjnego, który pośredniczy pomiędzy oprogramowaniem systemu sterowania a sprzętem sterownika PLC oraz oprogramowania systemu sterowania, czyli części algorytmu działania systemu przeznaczonego dla danego urządzenia wykonawczego. Bezpieczeństwo oprogramowania systemu sterowania oparte jest na odpowiednich narzędziach programistycznych (narzędziach do programowania, kompilatorach), językach programowania (wybór języka

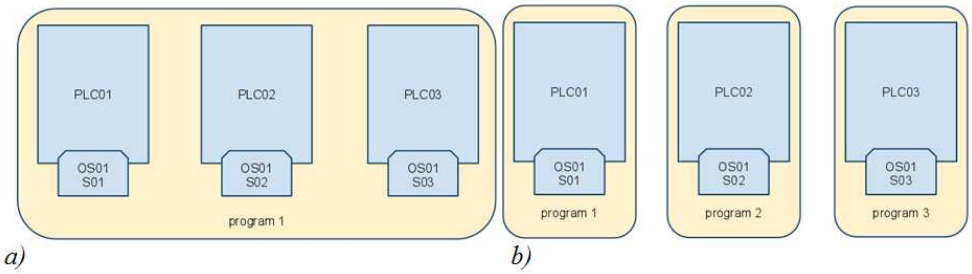
[1] Politechnika Śląska, Wydział Transportu; 40-019 Katowice; ul. Krasińskiego 8. Tel/Fax: + 48 32 603-41-36, e-mail: jerzy.mikulski@polsl.pl, szymon.surma@polsl.pl

oraz ograniczenia w ramach składni języka), technikach nadmiarowych kodów programu (programowanie defensywne, weryfikacja poprawnej kolejności wykonywania czynności) oraz doświadczeniu programisty (posługiwanie się językiem programowania, tworzenie dokumentacji oraz umiejętność wykrywania błędów).

Bezpieczeństwo systemu operacyjnego może zależeć od producenta sprzętu lub od programisty systemu sterowania, co wynika ze źródła pochodzenia systemu operacyjnego. Najczęściej, w implementacji sterowników PLC, korzysta się z systemu operacyjnego dostarczonego przez producenta, jednak dla pewnych zastosowań lepszym rozwiązaniem jest stworzenie własnego systemu operacyjnego (wymaga to posiadania pełnej dokumentacji sprzętu).

3. SPOSOBY BUDOWY OPROGRAMOWANIA

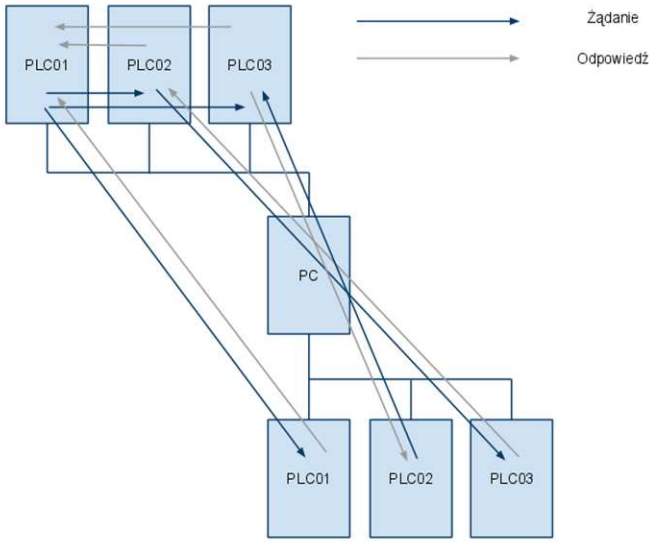
Dostępne narzędzia programistyczne pozwalają na tworzenie oprogramowania dla poszczególnych elementów systemu, jak i oprogramowywanie systemu sterowania jako całości. Podstawową różnicą pomiędzy tymi dwoma podejściami do programowania jest integracja poszczególnych urządzeń (sterowników PLC) z systemem i integracja warstw systemu. Proces budowy oprogramowania wymaga stworzenia pełnej dokumentacji opisującej wymianę informacji pomiędzy sterownikami i aplikację warstwy sieciowej do każdego sterownika z osobna.



Rys.1. Różnica pomiędzy tworzeniem oprogramowania w środowisku zintegrowanym (a) oraz w narzędziach do oprogramowywania poszczególnych elementów (sterowników PLC) (b)

Źródło: [Opracowanie własne]

Wykorzystanie w procesie budowy oprogramowania techniki oprogramowywania poszczególnych elementów systemu wymaga nieustannego weryfikowania założeń z wykonywanym programem. Wykorzystanie zintegrowanego środowiska programistycznego dla całości systemu, oparte na integracji warstwy sieciowej i programowej, nie pozwoli na popełnienie błędu w nazwie, typie lub przeznaczeniu zmiennej. Integracja warstw systemu nie tylko ułatwia zarządzanie zmiennymi i weryfikację poprawności wykorzystania, ale pozwala na śledzenie wymiany informacji w systemie jako całości.



Rys.2. Komunikacja w systemie scentralizowanym
Źródło: [Opracowanie własne]

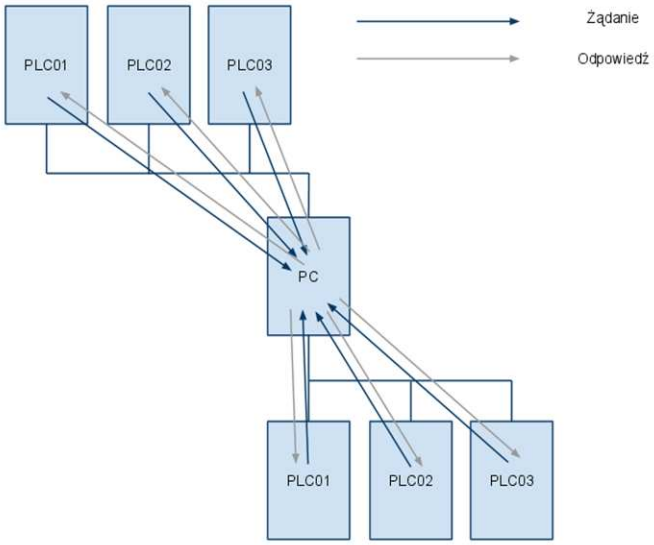
Informacje w systemie mogą pochodzić z kilku źródeł, z wejść i wyjść sterowników lub urządzeń wykonawczych lub mogą wynikać z działania systemu nadrzędnego (decyzyjnej warstwy sterowania). Rozpływ danych w ramach systemu może również być zorganizowany na różne sposoby:

- przekazywanie informacji opiera się o jednostkę centralną, która będzie gromadziła dane ze wszystkich urządzeń znajdujących się w systemie, a następnie „udzielała" tych informacji urządzeniom żądającym informacji.
- przekazywanie informacji polega na udostępnianiu informacji przez każde z urządzeń z osobna, czyli „udzielanie" informacji na żądanie.

Różnica pomiędzy tymi dwoma sposobami wymiany danych odzwierciedla jednocześnie sposoby organizacji samego systemu sterowania, czyli pracy systemu w wersji scentralizowanej i rozproszonej.

4. CENTRALIZACJA I DECENTRALIZACJA PROCESU STEROWANIA

System scentralizowany opiera się na jednym komputerze (sterowniku), który pełni funkcję nadrzędnego (podejmującego decyzje) i steruje pozostałymi elementami systemu. Taka konfiguracja pozwala na łatwe zarządzanie całym systemem i określenie funkcji poszczególnych jego elementów (urządzenia wykonawcze). Nie jest jednak optymalna z punktu widzenia transmisji danych, ponieważ agregować będzie duże przepływy w zakresie komunikacji ze sterownikiem centralnym. Wadą jest również podatność na uszkodzenia - usterka sterownika centralnego będzie skutkowała wyłączeniem całego systemu.



Rys.3. Komunikacja w systemie rozproszonym
Źródło: [Opracowanie własne]

System rozporoszony pozwala na umieszczenie fragmentów kodu nadrzędnego w poszczególnych sterownikach. Zmniejsza to obciążenie sieci - nie wszystkie dane muszą być przekazywane do centralnego sterownika. Pozwala również na zmniejszenie zagrożenia wyłączenia całego systemu sterowania z powodu usterki jednego sterownika.

Wykorzystanie cech przemawiających za danym rozwiązaniem wiąże się z wielkością docelowego systemu sterowania. Rozległe systemy sterowania będą bardziej podatne na decentralizacje niż małe systemy skupione. Jednocześnie należy brać pod uwagę bezpieczeństwo procesu sterowania oraz łatwość dowodzenia bezpieczeństwa. Złożoność problemu bezpieczeństwa w systemach sterowania prowadzi do wniosku, że należy tworzyć systemy oprogramowywane jak najbardziej zintegrowanymi programami. Pozwoli to na tworzenie systemów sterowania, w których poszczególne warstwy będą się przenikać, tworząc czytelny dla twórcy graf procesu sterowania, przy jednoczesnej głębokiej integracji warstwy sieciowej.

5. SYSTEMY STEROWANIA RUCHEM KOLEJOWYM

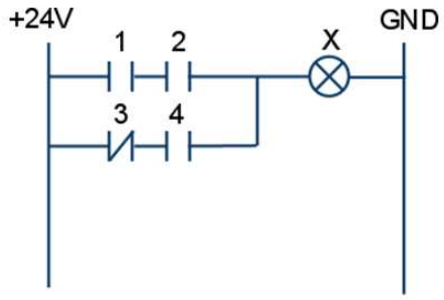
Systemy SRK bazujące na rozwiązaniach mikroprocesorowych mają coraz większy udział w urządzeniach sterowania ruchem. Aplikacje kolejowe sterowników PLC w Polsce nadal stanowią mały udział w ogólnej liczbie urządzeń mikroprocesorowych, głównie ze względu na restrykcyjne wymagania stawiane systemom SRK. Problematyczne jest nie tyle zachowanie poziomu integralności bezpieczeństwa (SIL) co jego udowodnienie. Aktualne zalecenia, które wykorzystuje się przy dopuszczaniu urządzeń do stosowania na liniach kolejowych bazują na sposobach określania poziomu bezpieczeństwa systemów przekaźnikowych i opartych na nich hybrydach. Rozwiązania oparte na sterownikach PLC różnią się od systemów przekaźnikowych nie tylko budową (urządzenia wykonawcze mogą

być przekaźnikami lub przekaźnikami elektronicznymi - opartymi na elementach optycznych lub półprzewodnikowych) lecz także zasadą spełniania zależności pomiędzy urządzeniami (np. zależności stacyjne).

Zasady bezpieczeństwa regulują wytyczne [10] wymuszające poziom bezpieczeństwa nie gorszy od urządzeń przekaźnikowych oraz normy europejskie [2,3,4]. Normy te określają, co prawda, poziomy bezpieczeństwa oraz sposoby ich osiągnięcia w przypadku oprogramowania, ale jednocześnie określają w odmienny do dotychczasowego sposób określania poziomu SIL - jako akceptowalne ryzyko wystąpienia usterki niebezpiecznej. Potencjalnie zbliżone określenia są od siebie różne, przede wszystkim z powodu różnic w sposobie postrzegania. W systemach przekaźnikowych sposób określania bezpieczeństwa wynikał wprost z wartości niezawodności poszczególnych elementów – przekaźników lub ich zestyków, jak w równaniu 1.

$$B = 1 - [(P_1 \cdot P_2 + P_3 \cdot P_4) \cdot P_X] \quad (1)$$

gdzie: B - bezpieczeństwo systemu,
P_1, P_2, P_3, P_3 - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia elementów
P_X - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia żarówki X.



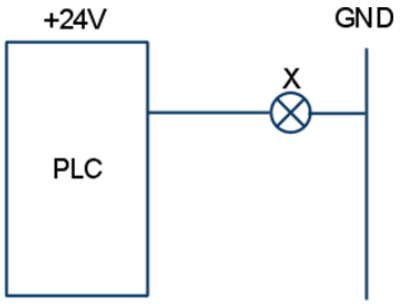
Rys.3. Model fragmentu systemu opisanego zależnością z równania 1
Źródło: [Opracowanie własne]

W przypadku systemów opartych na PLC każde z prawdopodobieństw nie jest zależne wyłącznie od sprzętu i obliczane wg. wzoru 1, ale również przemnażane przez wartość prawdopodobieństwa popełnienia błędu w programowaniu, które uzależnione jest od wielu czynników. Na wartość prawdopodobieństwa popełnienia błędu w oprogramowaniu wpływ ma narzędzie do programowania (w tym kompilator), doświadczenie oraz warunki psychofizyczne (samopoczucie) programisty i weryfikatora oraz dopasowanie systemu operacyjnego do sprzętu.

$$B = 1 - [(P_{1a} \cdot P_{1b} \cdot P_{1c})P_{OS} \cdot P_X] \quad (2)$$

gdzie: B- bezpieczeństwo systemu,
P_{1a} - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia sterownika PLC1,
P_{1b} - prawdopodobieństwo popełnienia błędu przez programistę 1,

P_{1c}- prawdopodobieństwo popełnienia błędu przez weryfikatora 1,
P_{OS} - prawdopodobieństwo niedopasowania systemu operacyjnego,
P_X - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia żarówki X.

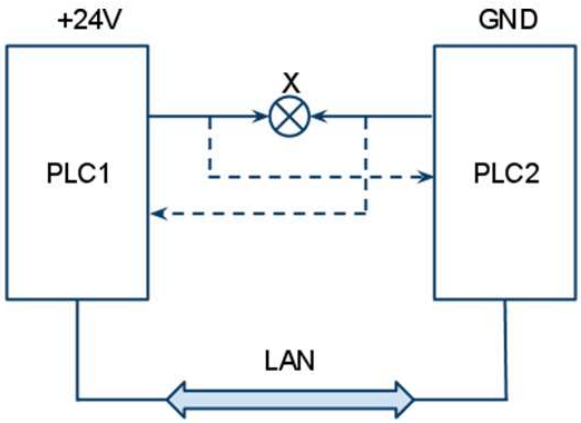


Rys. 4. Model fragmentu systemu opisanego równaniem 2.
Źródło: [Opracowanie własne]

Wartości prawdopodobieństwa dla systemu operacyjnego oraz narzędzia programistycznego można przyjąć za wartość stałą dla systemu opartego o jeden rodzaj sprzętu i dla którego oprogramowanie zostało stworzone w zintegrowanym narzędziu programistycznym. Jeśli oprogramowanie było tworzone dla każdego z elementów systemu w postaci oddzielnego projektu, należy wzór 2 uzupełnić o wartości prawdopodobieństwa popełnienia błędu przez programistów dla projektu PLC1 i projektu PLC2, jak również prawdopodobieństwa popełnienia błędu w integracji warstwy sieciowej oraz integracji zmiennych. Wartości te zostały uwzględnione we wzorze 3 oraz zilustrowane na rys. 5.

$$B = 1 - [(P_{1a} \cdot P_{1b} \cdot P_{1c} + P_{2a} \cdot P_{2b} \cdot P_{2c})P_{OS} \cdot P_{NLI} \cdot P_X] \quad (3)$$

gdzie: B - bezpieczeństwo systemu,
P_{1a} - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia sterownika PLC1,
P_{1b} - prawdopodobieństwo popełnienia błędu przez pierwszego programistę,
P_{1c} - prawdopodobieństwo popełnienia błędu przez pierwszego weryfikatora,
P_{2a} - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia sterownika PLC2,
P_{2b} - prawdopodobieństwo popełnienia błędu przez drugiego programistę,
P_{2c} - prawdopodobieństwo popełnienia błędu przez drugiego weryfikatora,
P_{OS} - prawdopodobieństwo niedopasowania systemu operacyjnego
P_{NLI} - prawdopodobieństwo niedopasowania warstwy sieciowej oraz braku integracji zmiennych,
P_X - prawdopodobieństwo wystąpienia niebezpiecznego uszkodzenia żarówki X.



Rys. 5. Model fragmentu systemu opisanego równaniem 3
Źródło: [Opracowanie własne]

Wartość niezawodności sterowników PLC jest określana jako MTBF (średni czas pomiędzy usterkami). MTBF jest jednym z czynników pozwalających określić wartość bezpieczeństwa systemu. Wartość współczynnika bezpieczeństwa przemysłowych systemów „bezpiecznych" nie przekraczające poziomu integralności bezpieczeństwa SIL3. Sposobem podwyższenia poziomu do wymaganego dla kolei SIL4 jest wprowadzenie redundancji sprzętowej oraz systemu 2 z 2 lub 2 z 3 wraz z głosowaniem sprzętowym. Przykład systemu 2 z 2 został zilustrowany na rys 5, gdzie żarówka X jest wysterowywana wartością dodatnią napięcia (+24V) z PLC1 i ujemną (GND) z PLC2. Zadziałanie żarówki X na rysunku 5 jest wynikiem sumy działania oprogramowania sterownika PLC1 oraz PLC2. Jakikolwiek błąd w oprogramowaniu PLC1 lub PLC2 dotyczący zmiennej, która wysterowywuje żarówkę X spowoduje pojawienie się niewłaściwego poziomu sygnału po którejś ze stron zasilania żarówki.

Jako dodatkowy sposób zapewniania bezpieczeństwa systemu zilustrowanego na rysunku 5, stosuje się krzyżową kontrolę poziomu sygnału po obu stronach żarówki. Oznacza to, że PLC1 odczytuje wartość poziomu sygnału wystawianego przez PLC2, a PLC2 odczytuje wartość poziomu sygnału wystawianego przez PLC1. Pozwala to na reakcje nie tylko w momencie niezadziałania żarówki X (gdy oczekiwane jest zadziałanie) ale również w czasie gdy oczekuje się niewysterowania żarówki. Taka organizacja systemu sterowania pozwala na dodatkową, oprócz programowej, weryfikację poprawności działania programu - gdy błąd będzie popełniony nie w kodzie programu nadrzędnego a w kodzie wykonawczym.

Realizacja sterowania zaprezentowana na rysunku 5 pozwala na eliminację błędów zarówno w przypadku systemów tworzonych w zintegrowanym środowisku programistycznym jak i dla każdego z elementów systemu indywidualnie.

6. WNIOSKI

Koncepcja budowy oprogramowania w zintegrowanym środowisku programistycznym przedstawiona w niniejszym referacie wskazuje na zasadność wykorzystania rozbudowanego i kompleksowego środowiska programistycznego do oprogramowywania

systemów sterowania ruchem kolejowym. Pozostanie przy aktualnie wykorzystywanym sposobie programowania będzie w dalszym ciągu pociągało za sobą zwiększone koszty tworzenia systemów sterowania. Zintegrowane środowisko programistyczne oferuje poza integracją warstw systemu sterowania również wsparcie w zakresie diagnostyki na etapie budowy systemu i oprogramowania, ale także na etapie eksploatacji systemu. Rozbudowane możliwości nowoczesnego oprogramowania pozwalają na ciągłe poszerzanie możliwości diagnostyczno eksploatacyjnych w przyszłości.

4. BIBLIOGRAFIA

[1] Dokumentacja systemu APROL 3.6 oraz rodziny sterowników X20 firmy Bernecker und Rainer (www.br-automation.com)
[2] Norma PN-EN 50126:2002
[3] Norma PN-EN 50128:2002
[4] Norma PN-EN 50129:2007
[5] Lewiński A.: *Problemy oprogramowania bezpiecznych systemów komputerowych transportu kolejowego*, Politechnika Radomska 2001
[6] Młyńczak J., Gorczyca P.: *Operation and diagnostic database of devices for railway traffic system*. Zeszyty Naukowe Politechniki Śląskiej, seria Transport nr 59, 2005, Gliwice, Politechnika Śląska 2005
[7] Mikulski J., Młyńczak J.: *Diagnosis of Point Machines with PLC Controller Application*, Signal + Draht, nr 12, 2003
[8] Młyńczak J.: *Using Databases in Switch Point Mechanism Diagnostics*, CCIS 104, Springer, 2010.
[9] Surma S., Młyńczak J.: *Algorytm automatycznego dostosowywania wskazań semafora*, Prace naukowe "TRANSPORT", Nr 1/26/2008, Politechnika Radomska 2008.
[10] Zakład sterownia ruchem kolejowym: *Wymagania bezpieczeństwa dla urządzeń sterowania ruchem kolejowym*, CNTK, Warszawa 1998.