

Projektowanie wymiany danych na przykładzie małego systemu sterowania i wizualizacji

Janusz Wrzuszczak

Wydział Elektrotechniki, Automatyki i Informatyki Politechniki Opolskiej

W artykule przedstawiono etapy projektowania rozproszonego systemu sterowania binarnego oraz jego wizualizacji, zrealizowanego z wykorzystaniem sterowników PLC Simatic 200 oraz monitora dotykowego Magelis XBTG 5330. Szczególną uwagę zwrócono na problemy związane z wymianą danych przez interfejsy ethernetowe z protokołami TCP/IP oraz MPI/PPI. Przedstawione zostały niektóre problemy związane z projektowaniem systemu sterowania rozproszonego, takie jak: synchronizacja węzłów sterujących systemu, konfiguracja logicznych kanałów komunikacyjnych oraz zdefiniowanie założeń harmonogramu wymiany danych dla potrzeb sterowania i wizualizacji. Pokazano przykładowe zmienne, istotne dla funkcjonowania systemu, wymieniane pomiędzy poszczególnymi węzłami. Zadanie projektowe zostało zilustrowane implementacją aplikacji sterowania, monitorowania i nadzoru w środowisku sterowników PLC Simatic S 7-200 [2], połączonych w segment sieci ethernetowej z wykorzystaniem przełącznika sieciowego oraz interfejsu RS-485 z protokołem Siemens PPI/MPI do komunikacji nadrzędnego węzła sterującego z panelem dotykowym Magelis XBTG 5330 [3]. Topologia systemu jest uzależniona jest w znacznym stopniu od możliwości dostępnych urządzeń komunikacji sieciowej i ich oprogramowania. Aplikacja miniSCADA w monitorze, pracująca pod systemem Vijeo Designer Runtime (ver. 4.6.0.3.3623) pozwala realizować wybrane funkcje takie jak: synchronizacja sterowników do pracy automatycznej, realizacja zadania bezkolizyjnego przejazdu przez ciąg trzech skrzyżowań tzw. „zielona fala”, sterowanie w trybie zgłoszeniowym, sterowanie w godzinach nocnych, sterowanie ręczne, diagnostyka pakietów wymiany danych. Zaprojektowany system zrealizowano w laboratorium automatyki przemysłowej Instytutu Automatyki i Informatyki Politechniki Opolskiej.

Słowa kluczowe: sterowanie rozproszone, SCADA, protokół sieciowy, logiczny sterownik programowalny, harmonogram wymiany danych

Funkcjonalność systemu rozproszonego jest określona poprzez architekturę systemu, w tym liczbę węzłów, ich interfejsy komunikacyjne, wykorzystywane protokoły komunikacyjne oraz rodzaj i charakter transmitowanej informacji. Szczególne znaczenie w funkcjonowaniu systemów rozproszonych mają długości komunikatów oraz ich wzajemne uwarunkowania czasowe, zwłaszcza w systemach czasu rzeczywistego [1]. Typowe zadania takie jak: pomiary, przetwarzanie danych i sterowanie muszą być uzupełnione o funkcje wytwarzania, rozdziału, transmisji i gromadzenia danych. Istotną rolę odgrywają ograniczenia wynikające z zastosowanego sprzętu i oprogramowania narzędziowego, nierzadko różnych generacji, pochodzących od różnych firm. Projektant w wielu przypadkach musi dostosować projekt do funkcjonujących już urządzeń, zebranych na zasadzie dołączenia różnych typów sprzętu lub podsystemów, wtedy głównym celem jest integracja funkcjonalna.

Architektura systemu

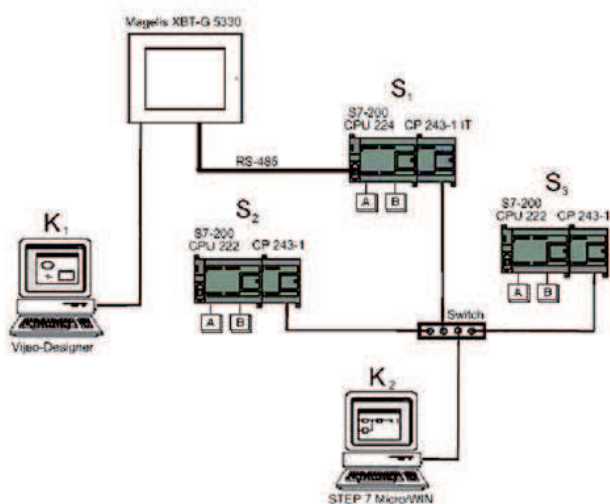
System rozproszony składa się z co najmniej kilku węzłów realizujących funkcje nadawania i odbioru danych z wykorzystaniem interfejsów komunikacyjnych typu punkt-punkt lub punkt-wielopunkt z wykorzystaniem różnorodnych protokołów sieciowych. Węzły te realizują zwykle podstawowe zadania dedykowane takie jak:

- dokonywanie pomiarów (czujniki lub przetworniki pomiarowe)

- przetwarzanie danych liczbowych na wielkości fizyczne (organy wykonawcze)
- realizacja algorytmów sterowania (regulatory)
- koncentratory danych (archiwizacja)
- wizualizacja.

Wiele różnych zadań może być zintegrowanych w jednym węźle lub mogą one być właściwością jedyną danego węzła. W zależności od topologii systemu zmieniają się charakterystyki czasowe wykorzystania poszczególnych rodzajów interfejsów komunikacyjnych oraz ich dostępności dla transferu poszczególnych komunikatów. Do wymiany danych w różnym stopniu można wykorzystywać wewnętrzne magistrale systemowe. W konsekwencji zadania realizowane przez poszczególne węzły mogą być spowalniane lub ulegać degradacji w wypadku wystąpienia znaczących opóźnień lub zagubienia komunikatów.

Topologia systemu rozproszonego składa się z węzłów S1, S2 i S3 zbudowanych ze sterowników Simatic S7-200, wyposażonych w interfejs sieciowy typu Ethernet CP 243-1 oraz połączonych za pomocą wejść/wyjść z makietami sygnalizacji świetlnej i czujników zgłoszeń pieszych na skrzyżowaniu dróg (rys. 1). Węzeł S1 pełni funkcję koncentratora danych transmitowanych interfejsem RS-485 z wykorzystaniem protokołu PPI/MPI do węzła zrealizowanego z wykorzystaniem monitora dotykowego. Niemożliwa jest bowiem komunikacja poprzez interfejs ethernetowy i protokół TCP/IP ze sterownikami Simatic S200. Komputery K₁ i K₂ pełnią funkcje programatorów i są wykorzystywane jedynie do załadunku aplikacji do monitora i sterowników PLC.



Rys. 1. Topologia rozproszonego systemu sterowania

Fig. 1. Topology of distributed control system

Urządzenia realizujące zadania w sieci należą do kategorii, takich jak urządzenia pomiarowe dostarczające telegramy zawierające wartość mierzonej wielkości w formacie binarnym lub zmiennoprzecinkowym. Urządzenia pełniące funkcje organów wykonawczych odbierają telegramy zawierające wartość ustawianej wielkości fizycznej w postaci kodu binarnego lub zmiennoprzecinkowego, albo wartość logiczną rozumianą jako rozkaz typu załącz/wyłącz. Bardziej zaawansowane technologicznie są węzły typu Programmable Logic Controller lub Programmable Automation Controller wyposażone w urządzenia wejścia/wyjścia, interfejs komunikacyjny oraz system operacyjny koordynujący wykonywanie zadań, obsługę interfejsów oraz obsługę błędów.

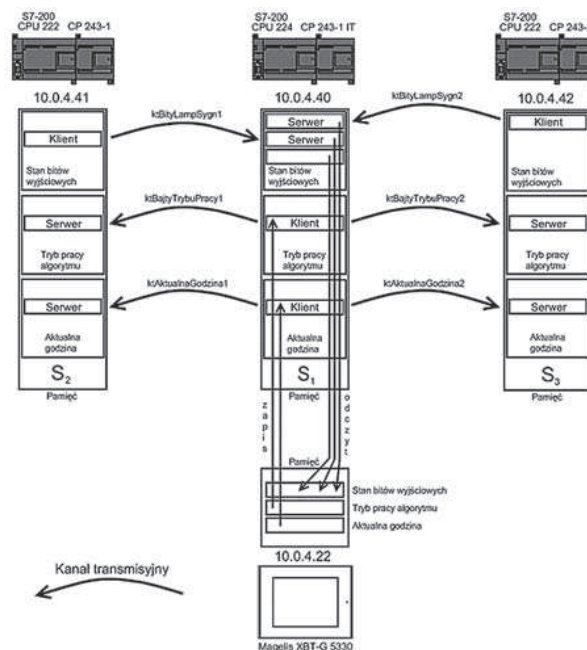
Zmienne systemowe i ich alokacja

Na początku etapu projektowania istotne jest zdefiniowanie zmiennych systemowych oraz ich alokacji w odpowiednich

Tab. 1. Wykaz wybranych zmiennych sterownika S1

Tab. 1. List of selected variables of the S1 PLC

Nazwa zmiennej	Adres	Komentarz
<i>kodBleduModuluEtherneto</i>	VW2004	Kody błędu Modułu Ethernetowego
<i>gotowoscKanalowTransmis</i>	VW2002	Gotowość kanałów transmisyjnych. Poszczególne bity reprezentują kanały transmisji
<i>gotowoscKanal4</i>	M0.4	Gotowość kanału na przesył trybu pracy do S3
<i>gotowoscKanal5</i>	M0.5	Gotowość kanału na przesył aktualnej godziny do S3
<i>gotowoscKanal1</i>	M0.6	Gotowość kanału na przesył trybu pracy do S1
<i>gotowoscKanal0</i>	M0.7	Gotowość kanału na przesył aktualnej godziny do S1
<i>awaria</i>	M2.0	Bit w stanie 1 oznacza awarię urządzenia
<i>trybDziennyBit</i>	V200.6	Ustawiony na 1 gdy ma być realizowany tryb dzienny
<i>trybNocnyBit</i>	V200.5	Ustawiony na 1 gdy ma być realizowany tryb nocny
<i>poOstCykluGotowosc</i>	M1.0	Ostatni takt cyklu się zakończył. System czeka na pozostałe
<i>polecenieSynchronizacji</i>	M1.2	Wydano polecenie, aby sterowniki się zsynchronizowały
<i>gotowoscS1</i>	M3.0	Sterownik S1 ukończył ostatni takt cyklu
<i>bitTrybDzienny</i>	V198.6	Ustawia na sterowanie w trybie dziennym
<i>zeruj</i>	SM0.1	Ustawiony przy pierwszym cyklu po załączeniu CPU w RUN lub po restarcie
<i>bitTrybAwaryjny</i>	V198.7	Ustawia na sterowanie w trybie awaryjnym



Rys. 2. Schemat wymiany danych między węzłami sieci

Fig. 2. Scheme of data exchange between network nodes

obszarach pamięci węzłów sieci. Szczególną rolę odgrywają te zmienne, które transmitowane są poprzez interfejsy sieciowe w formie telegramów zawierających komunikaty oraz zmienne systemowe. Wymieniane są one cyklicznie lub asynchronicznie, w przypadku kiedy generowane są sporadycznie. Nowoczesne protokoły wykorzystują mechanizm arbitra magistrali do realizacji sekwencji cykli w makrocyklach pozwalając zaprojektować ruch asynchroniczny możliwie równomierny w czasie [5]. W tab. 1. pokazano wybrane zmienne systemowe wykorzystywane w sterowniku S1, koordynującym wymianę danych w całym systemie. Zmienne systemowe zostały nazwane zgodnie z zaleceniami tzw. notacji węgierskiej.

W systemach prostszych opracowanie harmonogramu wymiany komunikatów oraz określenie okresów ich generowania, a także alokacji buforów wymiany danych dokonywane jest przez projektanta systemu.

Konfigurowanie kanałów ethernetowych zrealizowano za pomocą programu MicroWIN Ethernet Wizard. Kanał wykorzystujący protokół MPI/PPI skonfigurowano programem Vijeo Designer (Magelis), natomiast od strony sterownika S1 zdefiniowano port komunikacyjny 0. Do komunikacji wykorzystano infrastrukturę sieci LAN w laboratorium automatyki przemysłowej z przełącznikiem ethernetowym [6].

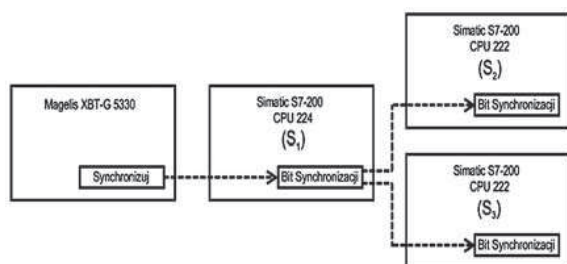
Na rys. 2. pokazano mechanizmy wymiany danych między węzłami sieci oparte na schemacie klient-serwer.

Aplikacje sterowania

Aplikacje sterowania zrealizowane zostały w językach programowania dostępnych w oprogramowaniu Step7 MicroWIN, Vijeo Designer oraz skryptów Javy. Rolę zarządcy

systemu pełni aplikacja w węźle monitora dotykowego. Poszczególne zadania sterowania zostały zaprojektowane z wykorzystaniem diagramów stanów. Zależności czasowe zaprojektowano na podstawie diagramów harmonogramowania. Najważniejsze zadania systemu to synchronizacja, sterowanie ruchem ulicznym na ciągu trzech skrzyżowań, oraz funkcje systemu SCADA.

Synchronizacja węzłów po załączeniu napięcia w systemach rozproszonych wymaga rozesłania telegramów zawierających znaczniki czasu w celu ustalenia jednolitego czasu w całym systemie. W zależności od typu i architektury zastosowanych sterowników, mogą one być wyposażone w moduły czasu rzeczywistego, bądź należy zastosować indywidualną obsługę czasu systemowego. W wielu przypadkach należy po komendzie synchronizacji doprowadzić algorytmy lokalne do osiągnięcia zdefiniowanych wcześniej stanów startowych, będących stanami gotowości do załączenia algorytmu rozproszonego. W przykładowym systemie sterowania (rys. 3) synchronizacja węzłów polega na zainicjowaniu procedur sterowania w sterownikach PLC oraz monitorze dotykowym. Potwierdzenia o osiągnięciu przez wszystkie węzły stanu gotowości dostarczane są do centralnego węzła sieci, jakim jest minisystem SCADA pracujący w trybie operatorskim lub inżynierskim.



Rys. 3. Synchronizacja węzłów po załączeniu napięcia

Fig. 3. Synchronizing nodes after power supply switching on

Obsługa awarii w systemie wymaga wdrożenia algorytmów lokalnych prowadzących do bezpiecznego doprowadzenia zadań do stanu początkowego lub całkowitego wyłączenia.

Testowanie systemu transmisji w sieci Ethernet przeprowadzono wykorzystując program *wireshark* [7] pozwalający prześledzić ruch w sieci. Wykorzystano w tym celu jeden z portów programowalnego przełącznika sieciowego z agregacją ruchu z portów pozostałych.

Podsumowanie

Projektowanie systemów rozproszonych wymaga starannego zaprogramowania wymiany danych pomiędzy węzłami w sieci. Dotyczy to częstości transmisji wybranych zmiennych systemowych, ich alokacji w pamięci węzłów, wyboru protokołów transmisji, a także narzędzi do programowania i testowania transmisji i funkcjonowania całego systemu.

Bibliografia

1. Sacha K.: *Systemy czasu rzeczywistego*. Wydawnictwo: OWPW 2006.
2. www.automation.siemens.com
3. www.schneider-electric.pl

4. Świerc T.: *Wizualizacja rozproszonego procesu z wykorzystaniem monitora dotykowego*. Politechnika Opolska 2010, praca dyplomowa pod kierunkiem J. Wrzuszczaka.
5. Grega W.: *Metody i algorytmy sterowania cyfrowego w układach scentralizowanych i rozproszonych*. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków 2004.
6. Wrzuszczak J., Grandek K.: *Sieci komputerowe w rozproszonych systemach automatyki*, PAK, 10/2006 s. 22–24.
7. www.wireshark.org

Design of data exchange in a distributed system. Case study of a small control system and visualization

In the paper have been presented the phases of design of the distributed binary control system and its visualization implemented with Siemens Simatic 200 PLCs and a Schneider touch monitor Magelis XBTG 5330. The special attention has been paid on the problems dealing with data transfer throughout the Ethernet interfaces with a TCP/IP and a Siemens MPI/PPI. There are pointed out some issues dealing with designing of DCS such as: synchronizing control nodes of the system, configuring logical communication channels and defining set up of data exchange schedule for control and visualizing tasks. Exemplary system variables being exchanged between various nodes and relevant for operation of the system has been depicted. A case study of the SCADA system has been discussed in the environment of Simatic S7-200 PLCs [2] connected in a network segment across a Ethernet switch and a RS485 interface supported by the Siemens MPI/PPI protocol adopted for communication of the PLC master control node with the Magelis XBTG 5330 [3] touch panel. The topology of the system substantially depends on the features of available network communication devices and their firmware. The mini-SCADA application in the touch monitor running under Vijeo Designer Runtime (ver. 4.6.0.3.3623) implements some control premises and algorithms i.e.: synchronizing PLCs prior the system transition to the automatic mode, road traffic control on the sequence of three road-crossing "a green wave", control on pedestrian request, operation in the night, manual operation, diagnostics of data exchange packets. The system designed after these rules has been implemented in the laboratory of the of industrial automation in the Institute of Automatic Control and Informatics at Opole University of Technology.

Keywords: distributed control, SCADA, network protocol, PLC, data exchange schedule

dr inż. Janusz Wrzuszczak

W roku 1998 uzyskał stopień doktora nauk technicznych. Pracuje jako adiunkt na Wydziale Elektrotechniki, Automatyki i Informatyki Politechniki Opolskiej. Jest autorem lub współautorem kilkudziesięciu publikacji naukowych. Zainteresowania naukowe to teoria sterowania, automatyzacja procesów przemysłowych, identyfikacja systemów, systemy rozproszone.



e-mail: j.wruszczak@po.opole.pl