

Jerzy MIKULSKI
Andrzej DUSZA
Michał PAWLICKI

ZASTOSOWANIE STEROWNIKÓW PLC W SYSTEMIE SKOORDYNOWANEJ SYGNALIZACJI ŚWIETLNEJ

Streszczenie. Artykuł traktuje o możliwościach i zastosowaniu sterowników programowalnych w systemie sterowania skoordynowaną sygnalizacją świetlną. w kolejnych rozdziałach opisane są:

- historia rozwoju i rodzaje sterowników PLC,
- produkty firmy Bernecker & Rainer – System 2003, struktura systemu i GDM.

Na zakończenie artykułu zaproponowano system sterowania skoordynowaną sygnalizacją świetlną przy użyciu sterowników B&R.

THE USE PROGRAMMABLE LOGIC CONTROLLERS IN LINKED TRAFFIC SIGNALS

Summary. The article treats of Programmable Logic Controllers and possibilities to use them in linked traffic signals system. In following chapters are described:

- history and kinds of PLC,
- Bernecker & Rainer products – System 2003, structure and Graphical Design Method.

At the end of the article authors offer linked lights system based on Bernecker & Rainer Programmable Logic Controllers.

1. WSTĘP

Sterowniki programowalne PLC są szeroko stosowane do sterowania różnego rodzaju urządzeniami, procesami przemysłowymi i maszynami. Są to komputery specjalizowane, które pod kontrolą systemu operacyjnego czasu rzeczywistego wykonują następujące zadania [1]:

- zbierają wartości pomiarów za pośrednictwem modułów wejściowych z analogowych i cyfrowych czujników oraz urządzeń pomiarowych,
- transmitują dane za pomocą łącz i modułów komunikacyjnych,
- wykonują programy aplikacyjne na podstawie przyjętych parametrów i uzyskanych danych o sterowanym procesie lub maszynie,

- generują sygnały sterujące zgodne z wynikami obliczeń tych programów i przekazują je poprzez moduły wyjściowe do elementów i urządzeń wykonawczych,
- realizują funkcje diagnostyki programowej i sprzętowej.

Rozwój sterowników został zapoczątkowany przez firmę General Motors w roku 1968 [1]. Sterowniki programowalne znalazły zastosowanie w przemyśle samochodowym zastępując układy przekątnikowe w urządzeniach sterowania sekwencyjnego. Następnie wprowadzono sterowniki wyposażone w kasety sterowania zdalnego, które umożliwiły monitorowanie i uaktualnianie dużej liczby punktów wejść / wyjść. W 1977 roku firma Allen – Bradley Corporation jako pierwsza zastosowała w sterownikach mikroprocesor 8080 z wykorzystaniem dodatkowego koprocatora dla operacji bitowych [1]. Rynek sterowników gwałtownie wzrósł, gdy w 1983 roku w ofercie kilku japońskich producentów pojawiły się małe sterowniki o sporych możliwościach funkcjonalnych, a przy tym dużo tańsze od oferowanych dotychczas.

W latach dziewięćdziesiątych, wraz z wykorzystaniem komputerów IBM PC do programowania sterowników PLC, znacznie rozwinęły się możliwości programowe i komunikacyjne sterowników pracujących niejednokrotnie w rozwiniętych, sieciowych systemach komputerowych. Ponadto nastąpiło gwałtowne powiększenie asortymentu modułów inteligentnych, wyposażonych we własne układy mikroprocesorowe, które realizują nieraz bardzo złożone algorytmy przetwarzania sygnałów i procedury sterowania oraz komunikacji.

Z czasem pojawiła się konieczność standaryzacji sterowników PLC, a w szczególności metod programowania. Wynikiem tego było wydanie w 1993 roku normy IEC 1131 [1] pod nazwą „Programmable Controllers” – „Sterowniki programowalne”.

Norma IEC 1131 zawiera informacje ogólne oraz standaryzuje sprzęt i wymagania testowe. W części dotyczącej języków programowania norma definiuje podstawowe pojęcia, zasady ogólne, model programowy i komunikacyjny oraz ujednoliconą koncepcję programowania PLC w językach tekstowych i graficznych. Pozostałe części dotyczą wytycznych dla użytkownika oraz standaryzacji w zakresie wymiany informacji.

Biorąc pod uwagę wszechstronność zastosowań sterowników programowalnych PLC ze względu na ich elastyczność konstrukcyjną (budowa modułowa) i programową, możliwe jest wdrożenie systemu sterowania skoordynowaną sygnalizacją świetlną opartego na sterownikach programowalnych PLC. Takie rozwiązanie pozwala na ujednoczenie stosowanego sprzętu, szybką rozbudowę systemu oraz zaimplementowanie dowolnego algorytmu optymalizacyjnego w celu koordynacji pracy sygnalizacji świetlnej.

2. STEROWNIKI PROGRAMOWALNE

Na rynku dostępne są sterowniki będące produktami takich firm, jak: GE Fanuc, Simatic Allan – Bradley, AEG Schneider Modicon. Pomimo wielu wykonawców zasady programowania oraz budowa tych urządzeń są zbliżone do siebie i określone w normie IEC 1131 [1]. W związku z tym obecne rozwiązania konstrukcyjne sterowników programowalnych umożliwiają ich dużą elastyczność pod względem konfiguracji sprzętowych. Wynika to z faktu, iż sterowniki PLC posiadają budowę modułową. Niezbędnym elementem konstrukcyjnym jest płyta łączeniowa (kaseta), do której podłączane są określone moduły. Podstawowymi modułami są : moduł zasilacza i jednostki centralnej CPU. Inne, najczęściej wykorzystywanymi, to [1]:

- moduł wejść i wyjść dwustanowowych,
- moduł wejść i wyjść analogowych,

- moduł szybkiego liczenia HSC do obsługi wejść dwustanowych zmieniających się z częstotliwością nie większą niż 100 kHz, pochodzących najczęściej z przetworników impulsowo – obrotowych,
- moduł pozycjonowania osi APM,
- moduł komunikacyjny z dwoma łączami szeregowymi RS-232 i RS-422/485 z określonymi protokołami komunikacyjnymi,
- moduł komunikacyjny sieci sterownikowych Genius GCM do podłączania sterowników w sieci lokalnej,
- moduł komunikacyjny sieci Ethernet, który stanowi łącze między sterownikami i siecią Ethernet TCP/IP LAN. Umożliwia on komunikację w sieci Ethernet poprzez media transmisyjne takie, jak : kable koncentryczne, kable skrętki lub kable światłowodowe.

Oczywiście, każdy z produktów danej firmy charakteryzuje się pewną specyfiką związaną np. ze środowiskiem programowania, rozwiązaniami konstrukcyjnymi oraz oferowanymi modułami i bibliotekami funkcyjnymi. Przykład mogą stanowić przedstawione w opracowaniu sterowniki firmy B&R (Bernecker & Rainer).

B&R oferuje trzy rodzaje sterowników: System 2003, System 2005 i System 2010. Użytkowanie i programowanie poszczególnych typów sterowników nie różni się, natomiast inna jest maksymalna liczba wejść i wyjść różnych typów (analogowe i cyfrowe), jaką są w stanie „obsłużyć” poszczególne wersje systemu. „Najuboższy” jest System 2003, a najbardziej rozbudowany System 2010 i systemy kombinowane, czyli wykorzystujące wszystkie rodzaje sterowników.

Inne różnice polegają na rodzaju bibliotek funkcyjnych, jakie oferuje dany system, oraz liczbie dostępnych rodzajów zadań, do których należą [2]: zadania standardowe, zadania wysokiej prędkości HS i zadania specjalne.

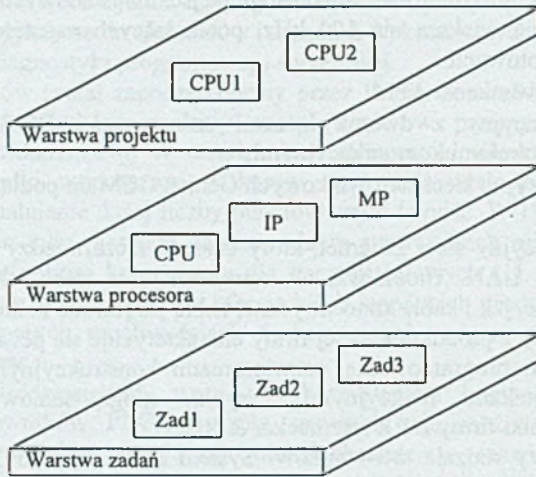
Natomiast od strony użytkownika podstawowym narzędziem programowania sterowników B&R jest System PG2000, który składa się z części zarządzającej i środowiska programowania. Część **zarządzająca** odpowiedzialna jest za otwieranie, kopiowanie oraz usuwanie projektów, natomiast **środowisko programowe** związane jest z samym programowaniem, kompilowaniem i ładowaniem kodu źródłowego do sterowników.

Dużym udogodnieniem w Systemie PG2000 jest wprowadzenie graficznej metody projektowej GDM, ułatwiającej podzielenie złożonego projektu na mniejsze części – warstwy, zadania i moduły – interpretowane graficznie. Dzięki temu narzędziu planowanie projektu, jego analiza oraz edycja stają się bardziej przyjazne dla użytkownika. Strukturę omawianej metody przedstawia rys. 1.

Warstwa projektu jest warstwą najwyższą, wyświetlaną zaraz po wywołaniu GDM. Przedstawione są na niej wszystkie jednostki centralne CPU (sterowniki), które obejmuje dany system.

Warstwa procesora jest opcjonalna i należy ją zainicjalizować, gdy jest to konieczne. W warstwie tej znajdują się symbole wszystkich modułów multiprocessorowych oraz modułów inteligentnych.

Warstwa zadań należy do najniższej położonych. Przedstawione są w niej w postaci odpowiednich symboli graficznych zadania oraz moduły programowe (biblioteki funkcyjne).



Rys. 1. Graficzna metoda projektowa GDM
Fig. 1. Graphical Design Method

Zadania złożone są z dwóch części [2]. Pierwsza z nich to program, druga zawiera deklarację zmiennych. Jednocześnie może być uruchomionych wiele zadań, które wymieniają informacje pomiędzy sobą. Zadania podzielone są w grupy nazwane klasami zadań. Takie ich pogrupowanie umożliwia ograniczenie wymiany informacji między zadaniami różnych klas oraz ustalenie priorytetu wykonania.

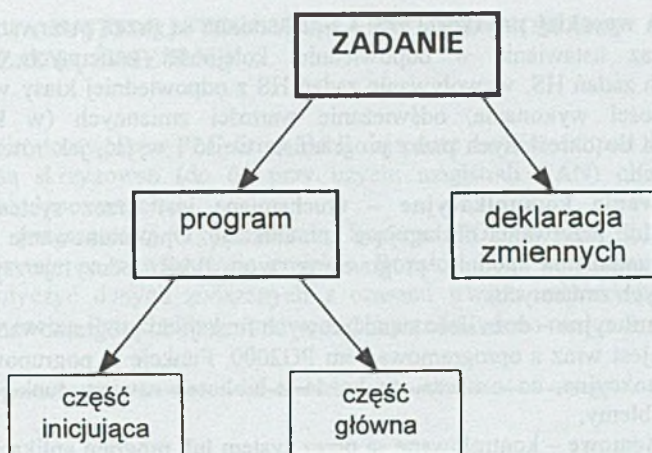
Program jest częścią składową zadania i zawiera dwie części: inicjującą i główną. Każda z nich może być napisana w różnych językach programowania.

Część inicjująca wykonywana jest tylko raz, podczas uruchamiania programu i służy do przypisywania wartości stałym oraz do ustawiania wartości początkowych określonych zmiennych. Ponadto w części inicjującej aktywuje się sterowniki programowe związane z obsługą sieci lokalnych, modemów itp.

Część główna wykonywana jest wielokrotnie, zaraz po zakończeniu części inicjującej. Oznacza to, że po wykonaniu części inicjującej oraz głównej następuje powrót do początku części głównej. Część inicjująca nie jest już wykonywana.

Deklaracja zmiennych następuje w oddzielnej części i polega na przypisaniu zmiennym występującym w programie fizycznych wejść, wyjść, adresów w pamięci oraz określenia ich typów i innych atrybutów.

Zależność między zadaniem, programem oraz ich elementami składowymi przedstawia rys. 2.



Rys. 2. Zadanie – elementy składowe, hierarchia

Fig. 2. Task – structure, hierarchy

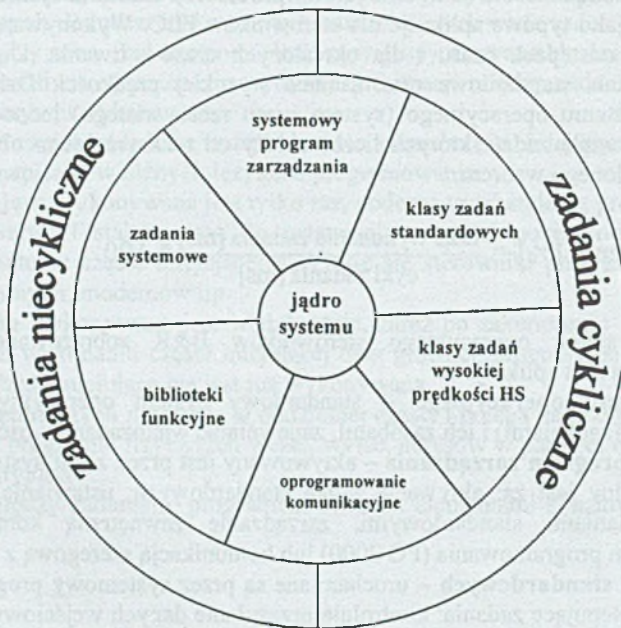
Zasadniczo zadania można podzielić na: zadania niecykliczne i zadania cykliczne. **Zadania niecykliczne** wykonywane są wtedy, gdy system operacyjny i zadania cykliczne nie wykorzystują czasu procesora (tzw. czas jałowy procesora). **Zadania cykliczne** kontrolowane są czasowo [3], jako typowe aplikacje dla sterowników PLC. Wykonywane są cyklicznie, w zdefiniowanych odstępach czasu i dla określonych czasów trwania. Do tej grupy zadań zalicza się zadania standardowe oraz zadania wysokiej prędkości. Dzięki odpowiedniej architekturze systemu operacyjnego (system czasu rzeczywistego) jednocześnie może być wykonywanych wiele zadań, których liczba zależy od maksymalnego obciążenia jednostki centralnej, określonego wzorem:

$$\text{Obciążenie [\%]} = \frac{\text{czas wykonania zadania [ms]} * 100}{\text{cykl zadania [ms]}}$$

Architekturę systemu operacyjnego sterowników B&R zobrazowano na rys. 3, a przedstawione na nim aplikacje to:

- **jądro systemu operacyjnego** – standardowy system operacyjny, który zarządza programami (zadaniami) i ich zasobami, zapewniając wielozadaniowość,
- **systemowy program zarządzania** – aktywowany jest przez zegar systemowy co 10 ms i odpowiedzialny jest za: aktywację zadań standardowych, ustawianie wejść i połączeń między zadaniami standardowymi, zarządzanie zewnętrzną komunikacją między środowiskiem programowania (PG 2000) lub komunikacją szeregową z panelami,
- **klasy zadań standardowych** – uruchamiane są przez systemowy program zarządzania i spełniają następujące zadania: kontrolują przesyłanie danych wejściowych i wyjściowych zadań standardowych, wywołują zadania standardowe w określonej dla nich kolejności wykonania, odświeżają wartości zmiennych (w każdym cyklu) przypisanych do określonych przez programistę wejść i wyjść, jak również zmiennych wewnętrznych,

- **klasy zadań wysokiej prędkości HS** – uruchamiane są przez przerwania i powodują: kontrolę oraz ustawianie w odpowiedniej kolejności zmiennych wejściowych i wyjściowych zadań HS, wywoływanie zadań HS z odpowiedniej klasy w określonej dla nich kolejności wykonania, odświeżanie wartości zmiennych (w każdym cyklu) przypisanych do określonych przez programistę wejść i wyjść, jak również zmiennych wewnętrznych,
- **oprogramowanie komunikacyjne** – uruchamiane jest przez systemowy program zarządzania lub przerwania obsługujące komunikację. Oprogramowanie komunikacyjne ładuje lub uaktualnia moduły programowe typu B&R, odczytuje/zapisuje wartości przetwarzanych zmiennych,
- **biblioteki funkcyjne** – duża ilość standardowych funkcji do programowania sterowników dostarczona jest wraz z oprogramowaniem PG2000. Funkcje te pogrupowane są w tzw. biblioteki funkcyjne, co oznacza, że każda z bibliotek zawiera funkcje rozwiązujące podobne problemy,
- **zadania systemowe** – kontrolowane są przez system lub program aplikacyjny i stanowią rozszerzenie systemu operacyjnego. Niektóre z nich przesyłane są automatycznie do sterownika, gdy wywołana jest określona funkcja. Inne zadania systemowe muszą być natomiast załadowane przez użytkownika wtedy, gdy jest to wymagane,
- **programy aplikacyjne** – zarządzane są przez system operacyjny. W związku z tym, że System B&R2000 jest wielozadaniowy, uruchomienie programów aplikacyjnych (zadań) następuje w tym samym czasie, a moc obliczeniowa procesora dzielona jest pomiędzy obsługiwane programy (zadania).



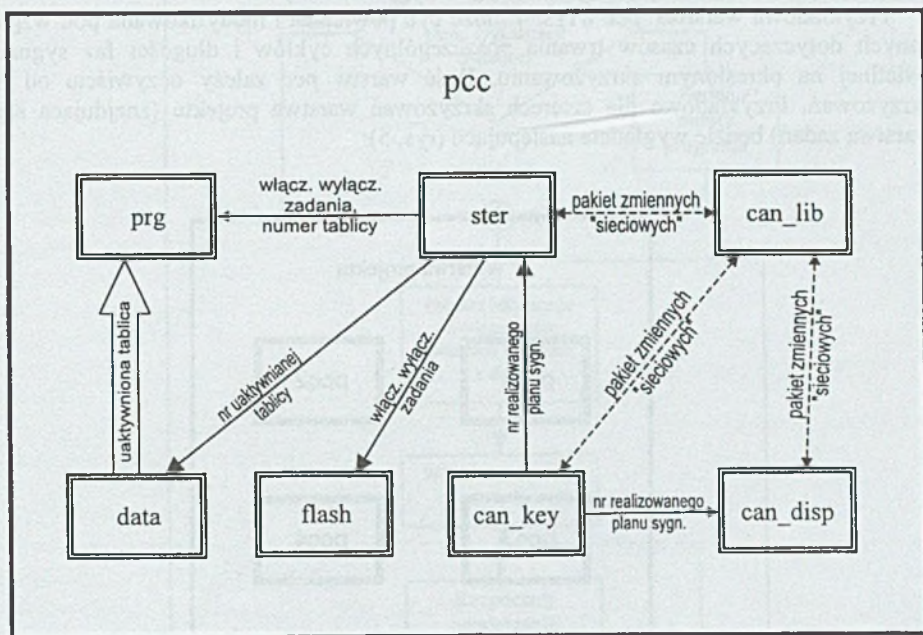
Rys. 3. Struktura systemu B&R 2000 [8]

Fig. 3. B&R 2000 system structure

3. KONCEPCJA SYSTEMU STEROWANIA SKOORDYNOWANĄ SYGNALIZACJĄ ŚWIETLNA

Zastosowanie sterowników PLC firmy B&R daje duże możliwości pod kątem sterowania określoną liczbą skrzyżowań (do 64 przy użyciu magistrali CAN) objętych systemem sygnalizacji skoordynowanych.

Od strony programowej wykorzystanie Systemu PG2000 oraz graficznej metody projektowej pozwala na standaryzację projektu. Różnice mogą występować w niektórych zadaniach i dotyczyć danych związanych z czasami trwania odpowiednich sygnałów i długościami faz ruchu. Przykładową warstwę zadań przedstawiono na rys. 4.



Rys. 4. Przykładowa warstwa zadań

Fig. 4. Task layer – example

- ⇨ – tablicę danych,
- – zmienną dowolnego typu,
- - - → – pakiet zmiennych związanych z biblioteką funkcyjną.

Do sterowania sygnalizacją świetlną służy program główny (zadanie) *prg*, który pobiera dane z zadania *data*, powoduje załączenie odpowiednich wyjść sterownika (włączenie świateł w komorach sygnalizatorów) oraz odmierza pobrane z zadania *data* czasy trwania poszczególnych sygnałów, faz ruchu i offsetów.

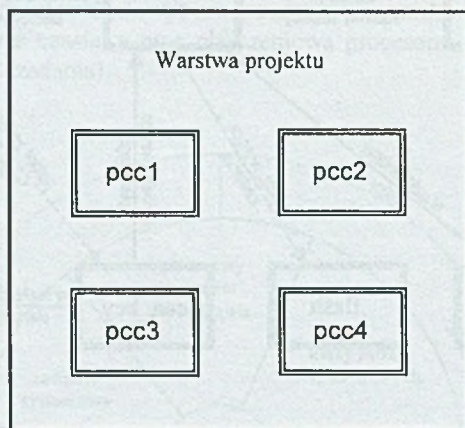
Zadanie *ster*, np. ze sterownika nadrzędnego pobiera dane określające rodzaj realizowanego w danej chwili planu sygnalizacji oraz odpowiedzialne jest za prawidłowe działanie sieci lokalnej. Proces ładowania sterowników „sieciowych” następuje w części

inicjującej programu i polega na wykorzystaniu odpowiednich funkcji uruchomieniowych z biblioteki systemowej *can_lib*, która jest „zaimportowana” do warstwy zadań. Biblioteka ta stanowi zbiór funkcji związanych z wysyłaniem i odbieraniem danych poprzez magistralę szeregową CAN.

Zadanie *flash* stanowi uzupełnienie programu głównego i realizuje program żółtych świateł migających na wszystkich wlotach. Zadanie to wykonywane jest albo jako jeden z planów sygnalizacji świetlnej (np. między godziną 22:00 a 6:00), albo w sytuacjach awaryjnych wynikających z nieprawidłowej pracy systemu.

Istnieje również możliwość podłączenia do systemu panelu (paneli) operatorskiego, którego sterowanie odbywa się w zadaniach *can_key* i *can_disp*. Zaproponowany panel może służyć do przeprowadzania czynności diagnostycznych poszczególnych sterowników w terenie, jak również zmiany programów sygnalizacji świetlnej.

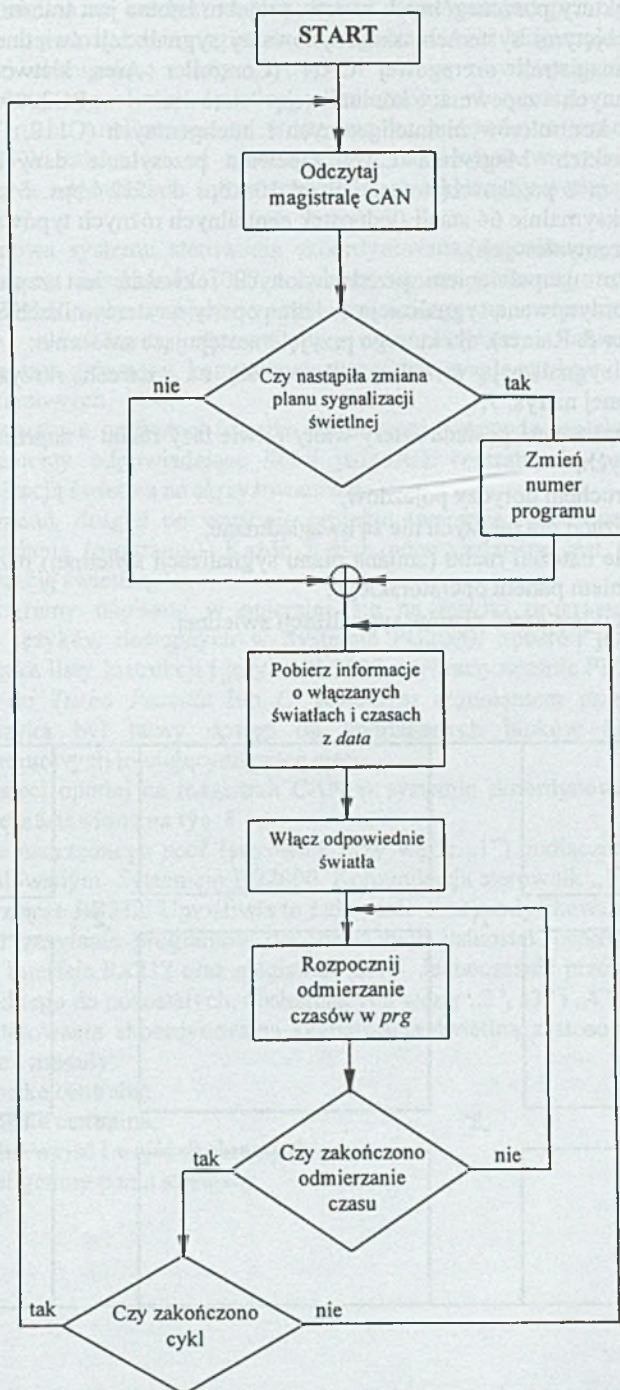
Przykładowa warstwa *pcc* z rys. 4 może być powielana i modyfikowana pod względem danych dotyczących czasów trwania poszczególnych cykli i długości faz sygnalizacji świetlnej na określonym skrzyżowaniu. Ilość warstw *pcc* zależy oczywiście od liczby skrzyżowań. Przykładowo dla czterech skrzyżowań warstwa projektu (znajdująca się nad warstwą zadań) będzie wyglądała następująco (rys. 5):



Rys. 5. Przykładowa warstwa projektu

Fig. 5. Project structure – example

Powracając do zadania *prg* z rys. 4, jego działanie można przedstawić za pomocą algorytmu, tak jak na rys. 6. Zadanie to jest „uniwersalne” i może być wykorzystane dla dowolnej sygnalizacji świetlnej.

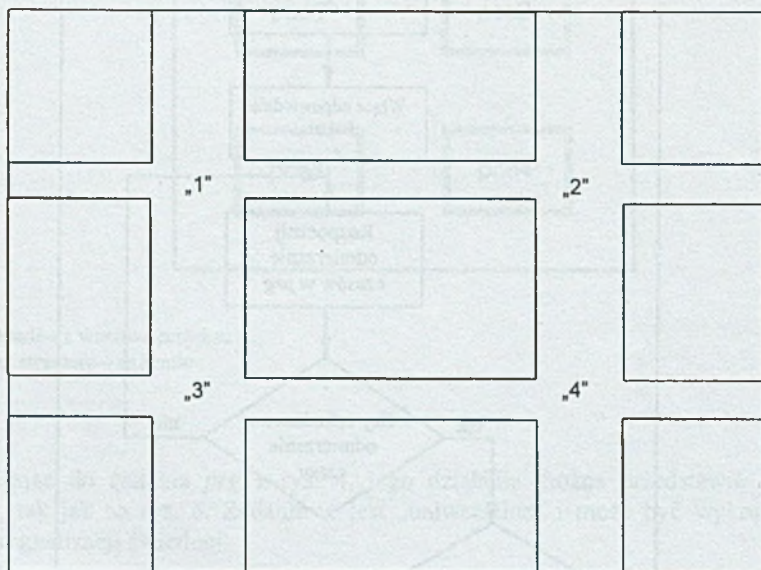


Rys. 6. Algorytm zadania prg
Fig. 6. Algorithm of task prg

Oprócz struktury poszczególnych warstw projektu istotna jest transmisja danych między sterownikami objętymi systemem skoordynowanej sygnalizacji świetlnej. Możliwe jest tu zastosowanie magistrali szeregowej CAN (Controller Area Network), która oprócz przesyłania danych zapewnia: komunikację sterownik – PG2000 oraz obsługę i programowanie kontrolerów nieinteligentnych i inteligentnych (C110, C130, C200, C300) paneli operatorskich. Magistrala CAN zapewnia przesyłanie danych na maksymalną odległość 1000 m z prędkością transmisji od 10 kbps do 512 kbps. Możliwe jest również podłączenie maksymalnie 64 stacji (jednostek centralnych różnych typów, paneli sterujących, komputerów przemysłowych).

Praktycznym uzupełnieniem przedstawionych rozważań jest zaproponowany system sterowania skoordynowaną sygnalizacją świetlną oparty na sterownikach Systemu 2003 firmy B&R (Bernecker & Rainer), dla którego przyjęto następujące założenia:

- sterowanie sygnalizacją świetlną odbywa się na czterech skrzyżowaniach w sieci przedstawionej na rys. 7,
- każde ze skrzyżowań posiada cztery wloty i dwie fazy ruchu – naprzemiennie dla wlotów przeciwnych,
- sterowanie ruchem dotyczy pojazdów,
- fazy sygnalizacji dla pieszych nie są uwzględniane,
- symulowanie natężeń ruchu (zmiana planu sygnalizacji świetlnej) przeprowadzane jest z wykorzystaniem panelu operatorskiego,
- do dyspozycji jest sześć planów sygnalizacji świetlnej.



Rys. 7. Model sieci ulic

Fig. 7. Road net model

Sygnalizacja świetlna na każdym ze skrzyżowań sterowana jest przez jeden sterownik lokalny połączony z pozostałymi magistralą szeregową CAN. Magistrala ta tworzy sieć lokalną. Sterowniki obsługujące węzły „2”, „3” i „4” są równouprawnione i realizują odpowiednie programy sygnalizacji świetlnej w zależności od sytuacji ruchowej w sieci. Wszystkie dane o ruchu analizowane są w sterowniku nadrzędnym umieszczonym przy skrzyżowaniu „1”. W zależności od natężeń ruchu na wlotach wszystkich skrzyżowań sterownik ten wysyła odpowiednie informacje do sieci lokalnej powodując zmianę planu sygnalizacji świetlnej.

Część programowa systemu sterowania skoordynowaną sygnalizacją świetlną została opracowana przy użyciu narzędzia PG2000, z wykorzystaniem graficznej metody projektowej GDM. Dzięki GDM cały projekt został podzielony na odpowiednie elementy przyporządkowane określonym sterownikom oraz elementy odpowiedzialne za: wykonywanie programu głównego, komunikację, przechowywanie struktur danych i inicjację sterowników programowych.

W pierwszej warstwie graficznej metody projektowej (warstwie projektu) umieszczone zostały cztery elementy odpowiadające ilości jednostek centralnych (pcc) sterowników sterujących sygnalizacją świetlną na skrzyżowaniach.

W warstwie zadań, drugiej po warstwie projektu (warstwa procesora jest pomijana), umieszczone są zadania (programy). Każde z nich odpowiedzialne jest za pewną część sterowania sygnalizacją świetlną.

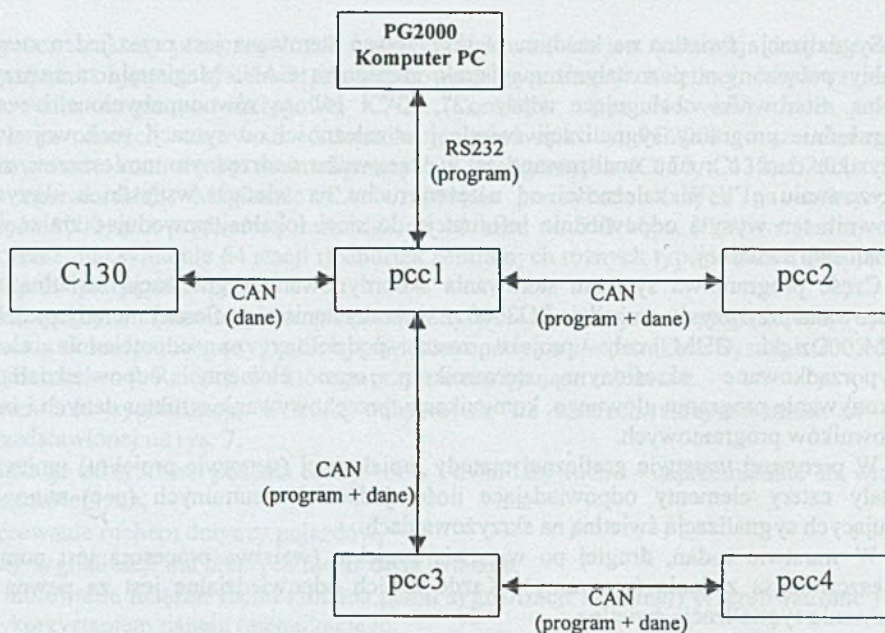
Wszystkie programy napisano w opierając się na języku programowania PL2000, jednym z trzech języków dostępnych w Systemie PG2000. Spośród języka schematów drabinkowych, języka listy instrukcji i języka PL2000, wybrano właśnie PL2000 ze względu na podobieństwo do *Turbo Pascala* lub *C*. Kolejnym argumentem przemawiającym za użyciem tego języka był łatwy dostęp do wymaganych bloków funkcyjnych oraz sterowników programowych inicjujących pracę sieci.

Konfiguracja sieci opartej na magistrali CAN w systemie skoordynowanej sygnalizacji świetlnej została przedstawiona na rys. 8.

Do sterownika nadrzędnego *pcc1* (sterownik przy węźle „1”) podłączony jest komputer klasy PC z zainstalowanym Systemem PG2000. Komunikacja sterownik „1” – komputer PC odbywa się przez złącze RS232. Umożliwia to ładowanie oraz modyfikowanie programów w tym sterowniku. Przesyłanie programów do pozostałych jednostek centralnych z PG2000 następuje poprzez interfejs RS232 oraz magistralę CAN. Jednocześnie przesyłane są dane ze sterownika nadrzędnego do pozostałych, obsługujących węzły „2”, „3” i „4”.

W systemie sterowania skoordynowaną sygnalizacją świetlną zastosowano następujące jednostki centralne i moduły:

- CP470 – jednostka centralna,
- CP474 – jednostka centralna,
- DM435 – moduł wyjść i wejść dyskretnych,
- C130 – nieinteligentny panel sterujący.



Rys. 8. Budowa sieci w systemie
Fig. 8. Net structure

Obsługa systemu sprowadza się do zadawania wybranych numerów planów sygnalizacji świetlnej przy użyciu panelu operatorskiego C130. Na wyświetlaczu ciekłokrystalicznym panelu podawane są równocześnie informacje o stanie systemu. Natomiast wizualizacja procesów zachodzących w systemie odbywa się na wykonanym modelu sieci z rys. 7.

4. PODSUMOWANIE

W systemach sterowania skoordynowaną sygnalizacją świetlną opartych na sterownikach programowalnych, w odróżnieniu od istniejących systemów, możliwe jest programowanie sterowników przy użyciu języka schematów drabinkowych, języka wysokiego poziomu (np. język PL 2000 lub C).

System zapewnia odpowiednie sterowanie ruchem w zależności od sytuacji w sieci. Wszystkie informacje o natężeniach ruchu analizowane są w sterowniku nadrzędnym, natomiast odpowiednie programy sygnalizacji w sterownikach lokalnych. Takie rozwiązanie zapewnia szybką rozbudowę systemu bez konieczności ingerowania w cały system, lecz tylko jego fragment. Również aktualizacja programów sygnalizacji świetlnej na danym skrzyżowaniu może przebiegać bez zakłócania pracy pozostałych sterowników. Taką korzyść daje zastosowanie sieci CAN. Sieć umożliwia połączenie jednostek centralnych sterowników na maksymalną odległość 1000m. Odległość ta jest wystarczająca ze względu na ograniczenie oddziaływania dopływów i odpływów bocznych w sieci ulic.

Budowa modułowa sterowników programowalnych umożliwia łatwą i elastyczną rozbudowę systemu sterowania skoordynowaną sygnalizacją świetlną pod kątem sprzętowym.

Dla systemu działającego w układzie rzeczywistym wymiana lub dołączenie dodatkowych modułów pozwala na zastosowanie pętli indukcyjnych, a tym samym lepsze dostosowanie planów sygnalizacji lub ich generowanie na bieżąco. Podłączenie pętli do systemu możliwe jest przy użyciu modułów analogowych wejść cyfrowych a przetwarzanie pobranych informacji bezpośrednio w programie. Zasilanie pętli indukcyjnych może natomiast odbywać się z szafy zasilającej.

Ponadto w systemie sterowania zapewnione są podstawowe wymogi bezpieczeństwa stawiane sygnalizacjom świetlnym. W systemie odbywa się kontrola sygnałów czerwonych. W przypadku przepalenia żarówki w komorze światła czerwonego następuje przełączenie sterownika w tryb pracy awaryjnej (żółty sygnał migający). Taka usterka nie zakłóca pracy pozostałych sterowników. Ze względu na budowę programu oraz format przechowywanych danych niemożliwe jest podanie przez sterownik sygnałów zielonych dla relacji kolizyjnych. Zbędne jest zatem wprowadzenie dodatkowego programu kontrolno-zabezpieczającego. W przypadku wystąpienia nieprawidłowości w działaniu sieci lokalnej CAN poszczególne sterowniki przechowują informacje o ostatnio realizowanym planie sygnalizacji świetlnej wykonując odpowiadający mu program. Również i w tym przypadku praca systemu nie zostaje zakłócona.

Zastosowany w systemie panel operatorski ma na celu zmianę planów sygnalizacji świetlnej. Jednak w rzeczywistym systemie może on być wykorzystany do modyfikowania programów w sterownikach lokalnych „w terenie” i przeprowadzania czynności diagnostycznych. Innym zastosowaniem panelu operatorskiego może być ułatwienie przejazdu przez sieć ulic pojazdom uprzywilejowanym. Panel ten pełniłby rolę lokalnego centrum sterowania i wpływałby na generowanie odpowiednich planów sygnalizacji, ułatwiając np. wyjazd karetki pogotowia ratunkowego. Znacznie lepszym rozwiązaniem ułatwiającym wyjazd pojazdów uprzywilejowanych mogłoby być zastosowanie na tych pojazdach radiomodemów komunikujących się z systemem.

W celu nadzoru pracy wszystkich sterowników możliwe jest przeprowadzenie wizualizacji procesów sterowania przy użyciu programu „In Touch”. Takie rozwiązanie umożliwiłoby stworzenie centrum sterowania zapewniającego szybką diagnostykę usterek oraz modyfikowanie programów sygnalizacji świetlnej z jednego miejsca.

Natomiast wykorzystanie algorytmu programu opartego na odpowiedniej metodzie optymalizacyjnej spowodowałoby lepsze dostosowanie planu sygnalizacji świetlnej do sytuacji ruchowej w sieci.

Literatura

1. Datka S., Suchorzewski M.: Inżynieria ruchu. WKiŁ, Warszawa 1999.
2. Programing Languages. B&R Industrial Automation – 1998.
3. System Software Reference Manual. B&R Industrial Automation – 1988.

Abstract

The B&R2000 PLC family is an automation system that is setting new standards in performance capability, functionality and operational safety.

The B&R 2003 system covers the complete range of applications from simple logic control to complex remote automation systems.

Deterministic multitasking operating systems allows the timing of the controller to be set optimally. The modular design of the operating system means this can be set up to match the application.

The conception of linked traffic signals shows, that it's possible to use Programmable Logic Controllers in traffic applications.