

**Sławomir JASZCZAK, Krzysztof MAŁECKI**WYDZIAŁ INFORMATYKI, ZACHODNIOPOMORSKI UNIwersYTET TECHNOLOGICZNY,  
ul. Żołnierska 49, 71-210 Szczecin**Synteza strukturalna algorytmu sterującego w sterownikach logicznych klasy micro**

Dr inż. Sławomir JASZCZAK



Od 1994, jako asystent na Wydziale Techniki Morskiej Politechniki Szczecińskiej, prowadzi badania w zakresie sterowania pojazdami głębinowymi z wykorzystaniem metod sztucznej inteligencji. Od 2002 zatrudniony na stanowisku adiunkta w Zakładzie Sztucznej Inteligencji. Bieżące zainteresowania wiążą się z implementacją złożonych algorytmów sterowania dyskretnego i cyfrowego na platformie wykonawczej PLC.

e-mail: [sjaszczak@wi.zut.edu.pl](mailto:sjaszczak@wi.zut.edu.pl)

Dr inż. Krzysztof MAŁECKI



Od 1997, jako asystent w Instytucie Informatyki Politechniki Szczecińskiej, prowadził badania dotyczące projektowania układów. Od 2001 zatrudniony na stanowisku adiunkta w ZUT w Szczecinie. Bieżące zainteresowania naukowe związane są z modelowaniem i symulacją układów złożonych, ze szczególnym uwzględnieniem ruchu drogowego.

e-mail: [kmalecki@wi.zut.edu.pl](mailto:kmalecki@wi.zut.edu.pl)**Streszczenie**

Sterowniki klasy micro mogą być z powodzeniem wykorzystywane do automatyzacji nieskomplikowanych instalacji przemysłowych, zawierających do kilkudziesięciu zmiennych procesowych o charakterze dwustanowym i/lub analogowym. W artykule przedstawiono metodykę tworzenia strukturalnego oprogramowania sterującego na przykładzie modelu laboratoryjnego układu sortowania materiału kolorowego, przy zastosowaniu sterownika klasy micro s7-200 CPU 224 z modułem rozszerzającym EM223.

**Słowa kluczowe:** synteza sprzętowo-programowa, PLC, programowanie strukturalne.

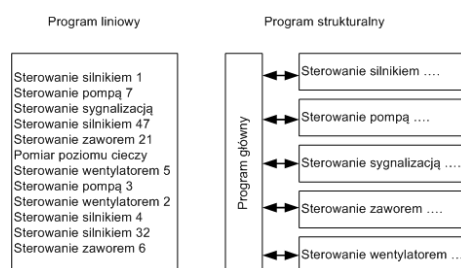
**Structural synthesis of the control algorithm in a micro programmable controller****Abstract**

Micro programmable controllers can be successfully used for automation of unsophisticated industrial installations, containing up to several dozen process variables of two-state and/or analog nature. This paper presents a methodology of creation of structural control software as an example of the laboratory model for sorting colored material, using a class of a micro controller S7-200 CPU 224 with an extended module EM223. The structural synthesis naturally allows taking into account the specific hardware configuration of the control object and functional requirements during the process of control software design. The functional requirements determine the structure of software, i.e. a developer may divide the program into smaller units, corresponding to different functions of the plant. At the beginning, an idea of the structural programming (Fig. 1) with an example in the micro PLC is described. In the next part a detailed example of the hardware and software synthesis, related to a real plant (Fig. 3) is given. A functional structure of the developed control system (Figs. 4, 5) has a great influence on the software structure (Fig. 6), what is described in details in that part. In the final part of this paper selected remarks about the structural implementation of the developed control algorithm in the s7-200 PLC are given and described.

**Keywords:** hardware and software synthesis, PLC, structured programming.

**1. Wprowadzenie**

W początkowej fazie rozwoju wszystkie programy sterujące, implementowane w pamięci sterowników PLC, miały strukturę liniową, jednak wraz z ewolucją języków programowania pojawiła się możliwość strukturalnego pisania programów. Metody projektowania strukturalnego dostarczają wskazówek do dekompozycji programów na mniejsze i łatwiejsze do zaprogramowania moduły. W metodzie tej programy mogą być tworzone na podstawie dobrze zdefiniowanych wymagań funkcjonalnych [5], precyzyjnie określających dopuszczalne rozwiązania problemu. Wykonanie tak napisanego programu polega na kolejnym wywołaniu podprogramów przez program główny. Na rys. 1 przedstawiono przykładowe rozwiązanie problemu sterowania pewnym systemem rzeczywistym zrealizowane za pomocą programu tworzonego liniowo i strukturalnie.



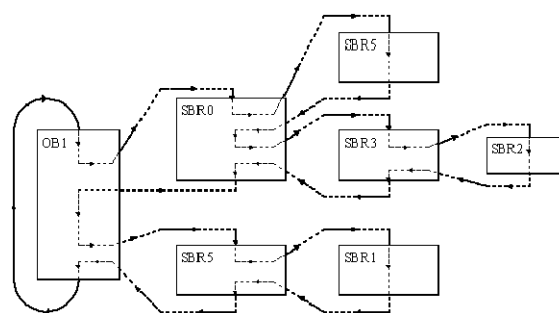
Rys. 1. Idea programowania liniowego i strukturalnego [7]  
Fig. 1. An idea of the linear and structural programming [7]

Niniejszy artykuł prezentuje syntezę strukturalną w odniesieniu do układu sortowania materiału kolorowego z wykorzystaniem sterownika PLC klasy micro w funkcji układu sterującego.

Zalety programowania strukturalnego [7] to:

- większa czytelność programu;
- ujednolicenie powtarzających się fragmentów programu – fragment powtarzający się może być napisany tylko raz w osobnym bloku i w miejscach jego występowania w programie wywołany jest tylko ten blok, z różnymi parametrami;
- prostsza realizacja, uruchamianie, testowanie i modyfikowanie programu – każdy z bloków programu może być tworzony i testowany niezależnie;
- możliwość tworzenia programu w grupie programistów, co skraca czas tworzenia nowego programu sterującego i modyfikacje.

Istotny jest w tym momencie fakt, iż moduły funkcjonalne programu sterującego są zdeterminowane poprzez określone moduły sprzętowe, realizujące zadane funkcjonalności urządzenia. W przypadku sterowników klasy micro, moduły funkcjonalne są przydzielane do podprogramów, wywołanych warunkowo lub bezwarunkowo tj. sytuacji, w której fragment kodu jest agregowany w większy moduł, dla poprawienia czytelności programu głównego.



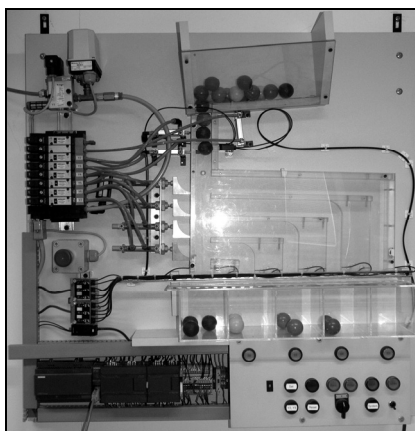
Rys. 2. Przykład programu strukturalnego w sterowniku S7-200. [7]  
Fig. 2. An example of the structural program in the S7-200 controller [7]

W sterowniku S7-200 program główny musi się znajdować w bloku OB1 – jest to blok nadrzędny w tym systemie. Podprogramy umieszczane są w blokach SBRxx i wykonywane są tylko wtedy, gdy bloki te są wywoływane. Program może być zagnieżdżony, tzn. podprogramy mogą wywoływać kolejne podprogramy, jednak liczba poziomów zagnieżdżenia jest ograniczona do 8. Do 64 jest też ograniczona ilość podprogramów. W systemie tym jest możliwa rekurencja, czyli wywołanie podprogramu przez samego siebie, można też wiele razy wywoływać ten sam podprogram z różnymi parametrami.

Na rysunku 2 przedstawiono przykładowy strukturalny program sterujący. Wywołanie poszczególnych bloków może być warunkowe, a więc pewne fragmenty programu nie muszą być zawsze wykonywane. Ma to wpływ na czas trwania cyklu PLC [2].

## 2. Przykład syntezy programu strukturalnego

Syntezę programu strukturalnego omówiono, wykorzystując rzeczywisty układ sortowania materiału kolorowego, przedstawiony na rys. 3.



Rys. 3. Widok układu sortowania materiału kolorowego [opracowanie własne]  
Fig. 3. The front view of the sorting control system [the authors' elaboration]

Z założenia zadaniem systemu sterującego jest posortowanie zmieszanego materiału kolorowego ze względu na kolor. Materiał stanowią różnokolorowe, gumowe kule o średnicy około 3,5 cm. Prędkość działania systemu jest określona przez prędkość spadania materiału do zasobników.

Wykorzystując założenia wstępne określono następujące funkcje systemu:

- sprawdzenie warunków wstępnych koniecznych do poprawnej pracy urządzenia;
- wykrywanie obecności materiału w zasobniku i rozpoznanie jego koloru;
- rozdzielanie materiału do osobnych zbiorników w zależności od jego koloru;
- komunikacja z operatorem - przyjmowanie poleceń;
- sygnalizacja stanów procesu i alarmów.

Schemat obrazujący poszczególne funkcje urządzenia przedstawiono na rysunku 4.



Rys. 4. Struktura funkcjonalna systemu sterowania [opracowanie własne]  
Fig. 4. The functional structure of the control system [the authors' elaboration]

Zgodnie ze schematem (rys. 4), operator ma możliwość ingerowania w każdą wyodrębnioną funkcję systemu, choć nie bierze bezpośredniego udziału w samym procesie sterowania procesem sortowania.

Szczegółowe funkcje urządzenia sortującego:

**załączenie systemu** - warunki wstępne konieczne do załączenia to:

- właściwie funkcjonujący układ sterowania (obecność zasilania, sterownik w trybie RUN, program w sterowniku uruchomiony),
- obecność sprężonego powietrza,
- nie załączony wyłącznik bezpieczeństwa.

**rozpoznanie koloru materiału:**

- musi odbywać się przez jedno urządzenie niezależnie od sterownika,
- urządzenie to musi być w łatwy sposób programowalne – tak, aby w każdej chwili można było wprowadzić materiał o różnym kolorze,
- urządzenie musi rozróżniać do 8 kolorów.

**rozdzielenie kolorowego materiału** do osobnych pojemników:

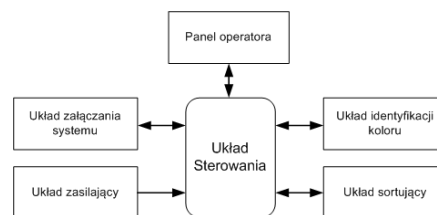
- w zależności od koloru materiał musi być kierowany do osobnych pojemników,
- materiał o kolorze różnym od zaprogramowanych kolorów ma być kierowany do osobnego pojemnika,
- system sterowania musi umożliwiać operatorowi wybór pojemnika, do którego ma trafiać materiał określonego koloru.

**komunikacja z operatorem** (czynności operatora):

- załączenie systemu – załączenie sprężonego powietrza,
- załączenie sortowania – uruchamia proces sortowania,
- załączenie programowania - wybór koloru i pojemnika - przypisanie materiału określonego koloru do określonego pojemnika,
- kasowanie alarmu – kontynuowanie sortowania.

**sygnalizacja stanu procesu:**

- system gotowy – oczekuje na załączenie sortowania lub programowania,
  - system w czasie pracy – sortowanie,
  - alarm – proces zatrzymany – alarm następuje, jeśli materiał skierowany do określonego pojemnika nie trafia tam mimo upływu określonego czasu, lub gdy brak materiału na wejściu.
- System można podzielić na szereg odrębnych, zależnych funkcjonalnie od siebie komponentów, przedstawionych na rys. 5.



Rys. 5. Moduły funkcjonalne systemu sortowania [opracowanie własne]  
Fig. 5. Functional modules of the sorting control system [the authors' elaboration]

**Układ sterowania** – konstrukcję oparto o sterownik, wyposażony w co najmniej 22 wejścia binarne i 14 wyjść binarnych, w którego programie ma być zawarty cały algorytm sterowania systemem.

**Układ załączania systemu** – załącza sprężone powietrze dopiero po pełnym starcie sterownika i kontroluje w sposób ciągły, czy ciśnienie powietrza przekracza określoną minimalną wartość – jeśli nie, odłącza powietrze i przesyła informację o tym do systemu sterującego. W skład układu wchodzi elektrozawór zasilania układu pneumatycznego, czujnik ciśnienia w układzie sprężonego powietrza, wyłącznik awaryjny oraz przycisk załączający system.

**Układ identyfikacji koloru** – ma za zadanie dostarczyć do sterownika informację o obecności oraz kolorze materiału na wejściu. Układ musi pracować niezależnie od sterownika i powinien rozróżniać 8 kolorów. Urządzenie to musi być w łatwy sposób programowalne – tak, aby w każdej chwili można było wprowadzić do pamięci materiał o kolorze innym od dotychczas rozróżnianych.

**Układ sortujący** ma za zadanie rozdzielić kolorowy materiał do różnych pojemników – zbudowany w oparciu o zespół rozdzielaczy i siłowników pneumatycznych. Rozdzielacze są sterowane bezpośrednio z wyjść binarnych sterownika i podłączone do instalacji pneumatycznej systemu. Siłowniki są sterowane powietrzem z wyjść odpowiednich rozdzielaczy. W skład układu wchodzi czujniki kontrolujące poprawną pracę układu – w przypadku gdy

materiał skierowany do określonego zbiornika nie trafia tam, sortowanie jest zatrzymywane i wystawiany jest alarm

**Panel operatora** – zawierający przyciski i przełączniki służące do sterowania procesem oraz lampki sygnalizacyjne informujące o stanie procesu. Zaplanowano następujące operacje możliwe do wykonania z panelu:

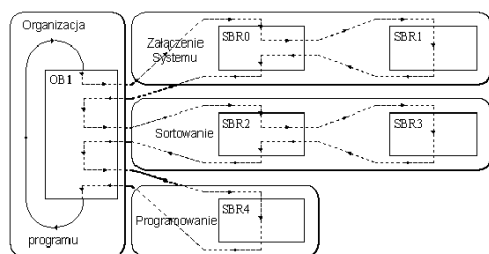
- załączenie systemu – sposób działania, wejścia i wyjścia opisane w punkcie *Układ załączania systemu*;
- załączenie i zatrzymanie sortowania;
- kasowanie alarmu – po skasowaniu sygnału alarmu sortowanie jest kontynuowane;
- programowanie – operator musi mieć możliwość przypisania materiału określonego koloru do określonego zbiornika. Programowaniu podlega również sam czujnik koloru. Po wejściu w tryb programowania sortowanie jest zatrzymywane.

**Układ zasilający** – powinien zapewnić standardowe napięcie zasilania 24VDC do sterownika oraz urządzeń zewnętrznych – czujników, rozdzielaczy pneumatycznych i elektrozaworów.

Wykorzystując moduły fizyczne podzielono program strukturalnie na cztery moduły funkcjonalne, które rozmieszczono w blokach SBR, zgodnie z rys. 6:

- Organizacja Programu,
- Załączanie Systemu,
- Sortowanie,
- Programowanie.

Umieszczono je w różnych blokach wywoływanych z bloku OB1, przy czym poszczególne bloki wywoływane są tylko wtedy, gdy zachodzi konieczność wykonania programów w nich umieszczonych. W danym momencie może być realizowany tylko jeden podprogram. O tym, który z podprogramów jest realizowany, decyduje operator – za pomocą przełącznika umieszczonego na pulpicie operatora.



Rys. 6. Rozmieszczenie programu w blokach SBR [opracowanie własne]  
Fig. 6. Location of the program parts in SBR blocks [the authors' elaboration]

Poszczególne moduły programowe tworzone były osobno i po ich przetestowaniu łączone były z modulem organizującym program, zawartym w bloku OB1.

Urządzenie sortujące może znajdować się w trzech trybach :

- urządzenie nie gotowe – czyli System nie załączony - obsługiwany powinien być moduł programowy Załączanie Systemu,
- urządzenie w trakcie sortowania – obsługiwany moduł Sortowanie,
- urządzenie w trybie programowania – obsługiwany moduł Programowanie.

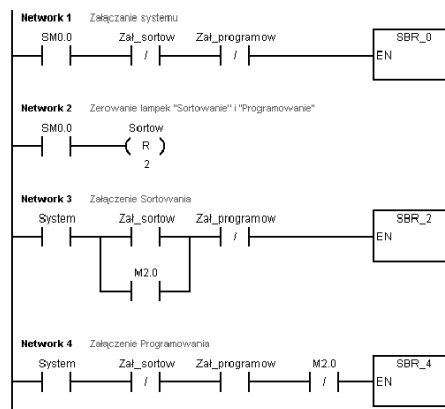
Za właściwą organizację i poprawną pracę całego programu odpowiedzialny jest moduł programowy Organizacja Programu, zawierający się w bloku OB1.

Zawartość bloku OB1, czyli bloku nadrzędnego w tym sterowniku, organizującego i spinającego pozostałe moduły programowe w jedną całość, przedstawiono na rys. 7.

W bloku SBR\_0 umieszczony jest program załączający System. Warunkiem wykonania tego podprogramu jest wyłączenie sortowania i programowania. Program umieszczony w SBR\_0 wywołuje kolejny blok – SBR\_1.

W bloku SBR\_2 znajduje się główny program urządzenia - program sortujący. Będzie on wywołany tylko wówczas, gdy załączony jest System, czyli informacja z bloku SBR\_0 o tym, że urządzenie jest gotowe do pracy, i nie jest załączone sortowanie.

Z bloku SBR\_2 jest wywoływany kolejny podprogram znajdujący się w bloku SBR\_3.



Rys. 7. Blok OB1 – Organizacja Programu [opracowanie własne]  
Fig. 7. OB1 block - Organization of the program [the authors' elaboration]

Warunkiem załączenia programowania, czyli programu znajdującego się w bloku SBR\_4, jest załączony system, nie załączone sortowanie oraz zakończony cykl sortowania.

Podsumowując można zauważyć, że podział oprogramowania na moduły wymaga wcześniejszego podziału na moduły sprzętowe, zgodnie z wymaganiami funkcjonalnymi. W takim przypadku warto przeprowadzić analizę funkcjonalną projektowanego urządzenia, wskazując na wstępie funkcję główną, a następnie dokonając podziału na funkcje składowe [8].

### 3. Wnioski

Głównym celem niniejszej pracy było udowodnienie, że podział oprogramowania sterującego na moduły, zgodnie z architekturą sprzętową maszyny przeznaczonej do automatyzacji, poprawia czytelność kodu, umożliwia wielokrotne wykorzystanie powtarzających się fragmentów programu, ułatwia realizację, uruchamianie, testowanie i modyfikację programu i daje możliwość pracy zespołowej, co zostało zaprezentowane na przykładzie opisywanego urządzenia do sortowania materiału kolorowego.

Według oceny autorów najistotniejsza wydaje się możliwość rozbudowy oprogramowania, bez konieczności modyfikowania istniejącej struktury. Dodanie funkcjonalności do maszyny, wiązałyby się z implementacją podprogramu i zapis do bloku SBR.

### 4. Literatura

- [1] Barczyk J.: Automatykacja procesów dyskretnych, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2003.
- [2] Broel – Plater B.: Sterowniki programowalne. Właściwości i zastosowania, Wydawnictwo Uczelniane Politechniki Szczecińskiej, Szczecin, 2000.
- [3] Legierski T., Kasprzyk J., Wyrwał J., Hajda J.: Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998.
- [4] S7-200 Programmable Controller System Manual, Siemens AG, Nürnberg, 2002.
- [5] Sacha K., Projektowanie oprogramowania systemów sterujących, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1999.
- [6] Szymiczek A.: Kurs Podstawowy S7, Centrum Szkoleniowe Systemów Automatyki INTEX, Gliwice, 2001.
- [7] Szyngiel P., Chmiel M.: Kurs S7-200 dla automatyków, Centrum Szkoleniowe Systemów Automatyki INTEX, Gliwice, 2002.
- [8] Urlich K.T., Eppinger S.D.: Product design and development, McGrawHill, New York, 2005.