

Mirosław CHMIEL
Instytut Elektroniki
Politechnika Śląska

O TWORZENIU PROGRAMÓW STEROWANIA DLA STEROWNIKÓW PROGRAMOWALNYCH

Streszczenie. W artykule skupiono się na niektórych elementach związanych z programowaniem (tworzeniem programu) sterowników programowalnych. Przede wszystkim zajęto się sposobami przedstawiania programów sterowania oraz możliwościami realizacji dostępu do danych procesowych i wpływu sposobu dostępu na tworzenie i wykonywanie programu sterowania. Poza tym przedstawiono krótki przegląd narzędzi do pisania programów i obserwacji procesu sterowania.

About programming of programmable controllers

Summary. In the presented paper attention is paid to some items related to programming (i.e. creating of a program) for Programmable Controllers. First of all ways of presenting control programs and abilities of accessing control data are discussed as well as the influence of data access procedures on creating and executing control programs. Moreover, a brief review of devices for service and monitoring of control process is given.

Autour du programmation des controleurs programmables

Résumé. Dans ce document, l'attention est portée sur des articles concernant la programmation des controleurs programmables. En premier lieu, l'article parle de toutes les manieres de presenter un programme de controle ainsi que les possibilites d'accéder aux donnees de controle, aussi bien que l'influence des procedures d'accès de donnees en creation et execution de programmes de controle. L'article presente ensuite une breve revue des composants de service et d'interceptions des processus de controle.

1. WSTĘP

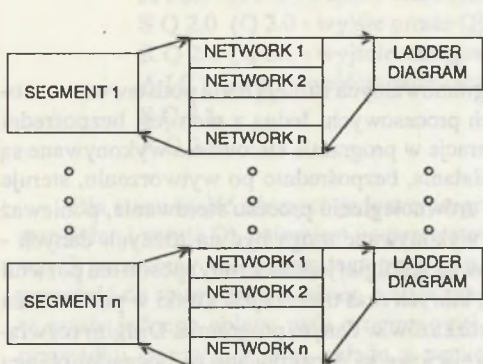
W sterownikach programowalnych bardzo ważnym elementem jest program sterowania, który powinien być skuteczny w działaniu, szybki oraz przejrzysty. Są to niezbędne cechy, jakimi powinien charakteryzować się dobrze napisany program. Dlatego też producenci sterowników, a właściwie ludzie tworzący do nich oprogramowanie, dają programistom różnego rodzaju mechanizmy, które ułatwiają pracę nad pisaniem programów. Są to przede wszystkim elementy związane z rozbijaniem programu na mniejsze części, programowanie strukturalne oraz możliwości decydowania o tym, kiedy dane wyjścia będą aktualizowane. Tym właśnie problemom

poświęcony został ten artykuł. Skupiono się przede wszystkim na trzech sterownikach, czy też rodzinach sterowników: Intelster PC4k [1], Modicon [2,3] i Simatic S5 [4,5,6].

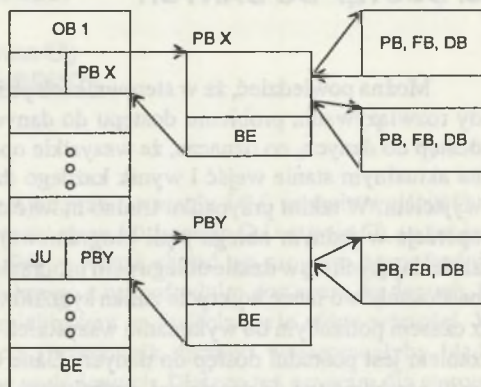
2. SPOSOBY PRZEDSTAWIANIA PROGRAMÓW STEROWANIA

Każdy producent sterowników tworzy swój własny program umożliwiający programowanie sterowników przez niego wytwarzanych. Programy te niekiedy dość znacznie się od siebie różnią, ale różnice te przeważnie tkwią w szczegółach, a ogólne założenia w większości przypadków pozostają takie same. Dla programowania sterowników Modicon 984 firmy Modicon AEG stworzono program, który nazywa się Modsoft. Na najniżym poziomie umożliwia on programowanie (tworzenie programu) za pomocą schematu stykowego. Mamy więc do dyspozycji styki (normalnie otwarty, normalnie zamknięty, impulsowe) oraz cewki (z pamięcią lub bez). Ponadto wszystkie bardziej skomplikowane operacje przedstawiane są w postaci bloków, do których wpisujemy odpowiednie parametry. W ten sposób uzyskujemy program w postaci schematu stykowego z blokami funkcyjnymi w postaci bloczków wywoływanych bezpośrednio z menu. Cały program sterowania przeważnie dzielony jest na mniejsze części, które mają za zadanie realizację pewnych elementów sterowania. W sterowniku Modicon 984 mamy możliwość pisania programów w segmentach, których może być maksymalnie 32. Kolejność oraz częstotliwość wykonywania danego segmentu można ustalić dowolnie w wykazie realizowanych segmentów (Segment Scheduler). Jeżeli jakiś segment jest krytyczny czasowo, można go wykonywać kilka razy w ciągu jednego obiegu pętli programowej. Każdy z segmentów z kolei składa się z sieci (network), w których umieszczamy program sterowania. Każda sieć ma wymiary 8 wierszy na 11 kolumn, a każde przecięcie kolumny z wierszem jest nazywane węzłem, w który wprowadzamy elementy. Przy czym są elementy jedno- lub wielowęzłowe w zależności od ilości parametrów, których dany blok wymaga.

Podobne możliwości daje język Step5 przeznaczony do programowania sterowników firmy Siemens - Simatic S5. W strukturze programu występuje blok organizacyjny (OB1), w którym piszemy część programu nazywaną sterowaniem nadrzędnym (strukturalnym). Blok OB1 spełnia podobną rolę co Segment Scheduler w języku Modsoft. Z bloku organizacyjnego można wywoływać następne bloki nazywane programowymi (PB), które zawierają programy użytkowe tworzące strukturę zgodną z technologicznymi i funkcjonalnymi warunkami sterowania. Z bloków programowych natomiast można wywoływać pozostałe bloki: funkcyjne (FB) - często powtarzające się elementy programowe, bloki danych (DB), czy też specjalne bloki sekwencyjne (SB). W blokach organizacyjnych istnieje także możliwość programowania liniowego. Liczba poziomów zagnieżdżenia programu może maksymalnie wynosić 16. W ramach bloku program piszemy w segmentach. Jeżeli chodzi natomiast o najniższy poziom pisania programu, program Step5 daje programiście większe pole manewru. Mianowicie można wybrać jedną z trzech metod programowania: LAD - schemat stykowy (podobny do tego z Modsofta), CSF - schemat logiczny oraz STL - lista instrukcji, przy czym pełny zestaw instrukcji dostępny jest tylko przy programowaniu metodą STL. Na rysunkach poniżej pokazano schematycznie, jak wygląda struktura programu dla sterowników Modicon 984 (rys.1) oraz Simatic S5 (rys.2).

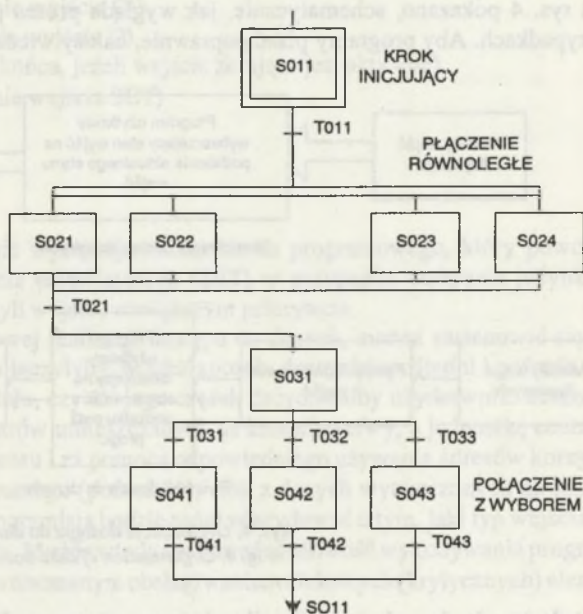


Rys. 1. Struktura programu Modsoft
Fig. 1. Structure of Modsoft program



Rys. 2. Struktura programu STEP - 5
Fig. 2. Structure of STEP - 5 program

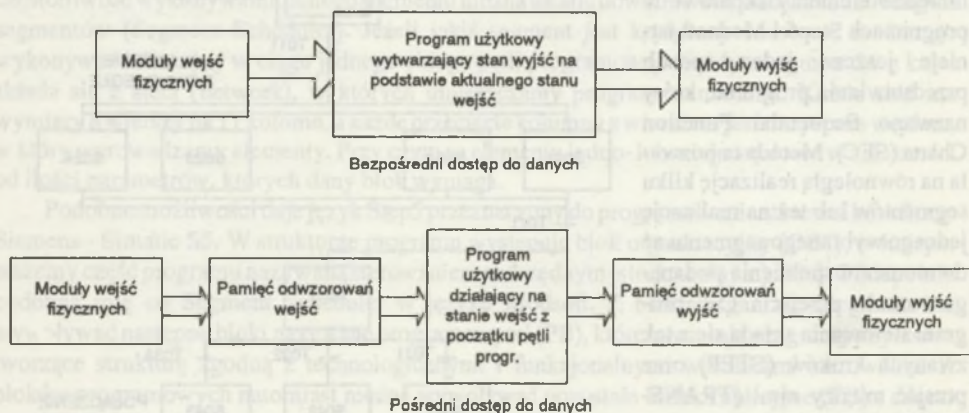
Oprócz przedstawionych sposobów podziału programu na mniejsze elementy składowe w programach Step5 i Modsoft istnieje jeszcze jeden sposób przedstawiania programu, który nazwano Sequential Function Charts (SFC). Metoda ta pozwala na równoległą realizację kilku segmentów lub też na realizację jednego wybranego segmentu aż do momentu spełnienia podanego warunku przejścia. Cały program sterowania składa się z tak zwanych kroków (STEP) oraz przejść między nimi (TRANSITION). Na rys. 3 podano przykład struktury SFC, która wygląda tak samo dla sterownika Modicon 984 jak dla Simatic S5. Pokazano na nim możliwości, jakie daje pisanie programów sterowania wykorzystujące tę właśnie metodę programowania. W każdym segmencie możemy stworzyć strukturę SFC, której elementy, z kolei, składają się z sieci (network). Jest to więc stopień pośredni między segmentem i siecią.



Rys. 3. Przykład programowania w SFC
Fig. 3. Example of SFC programming

3. DOSTĘP DO DANYCH

Można powiedzieć, że w sterownikach programowalnych istnieją dwie podstawowe metody rozwiązywania problemu dostępu do danych procesowych. Jedną z nich jest bezpośredni dostęp do danych, co oznacza, że wszystkie operacje w programie sterowania wykonywane są na aktualnym stanie wejść i wynik każdego działania, bezpośrednio po wytworzeniu, steruje wyjściem. W takim przypadku trudno mówić o zrównolegleniu procesu sterowania, ponieważ operacje w jednym obiegu pętli programowej wykonywane mogą być na różnych danych - zmieniających się w czasie obiegu pętli programowej. Z drugiej jednak strony sposób ten pozwoli na stosunkowo łatwe wykrycie zmian sygnałów, których czas trwania jest krótki w porównaniu z czasem potrzebnym do wykonaniu wszystkich rozkazów w danym programie. Drugim rozwiązaniem jest pośredni dostęp do danych. Dane wejściowe są odczytywane na początku obiegu pętli programowej do specjalnej, dodatkowej pamięci odwzorowań wejść. Następnie wykonywany jest cały program - instrukcja po instrukcji - a wyniki wykonywanych operacji przesyłane są do drugiej części pamięci dodatkowej, którą nazywa się pamięcią odwzorowań wyjść. Dopiero po zakończeniu realizacji programu wyniki wysyłane są na zewnątrz - do układów wyjściowych. Na rys. 4 pokazano, schematycznie, jak wygląda proces przekazywania danych w obydwu przypadkach. Aby programy pisać poprawnie, należy wiedzieć, w jaki sposób zorganizowany



Rys. 4. Organizacja dostępu do danych
Fig. 4. Organization of data access

jest dostęp do danych w sterowniku, którego używamy. Poniżej podano przykład realizacji funkcji asynchronicznego przerzutnika rs zrealizowanego dla obydwu sposobów dostępu do danych. Przykładem sterownika, w którym dostęp do danych jest bezpośredni, jest sterownik Intelster PC4k, a sterownik z dostępem pośrednim to na przykład Simatic S5. Zrealizowane przerzutniki to układy z aktywną jedynką na wyjściu, o dominującym zerowaniu, tzn. przy podaniu dwóch jedynek na wejścia, na wyjściu Q uzyskamy zawsze 0. Dla sterownika Simatic program można zrealizować następująco:

```

A I 0.0 (I 0.0 - wejście ustawiania SET)
S Q 2.0 (Q 2.0 - wyjście proste Q)
R Q 2.1 (Q 2.1 - wyjście zanegowane Q)
A I 0.1 (I 0.1 - wejście zerowania RESET)
S Q 2.1
R Q 2.0

```

Dla stanu wejść równych jedynie program po przetestowaniu I 0.0 ustawia wyjście Q w stan jeden i zeruje Q, natomiast po przetestowaniu stanu I 0.1 zeruje Q i ustawia Q, to też jest wynik ostateczny wykonania tego programu. Gdyby jednak chcieć ten program bezpośrednio przenieść do sterownika Intelster, a więc pracującego z bezpośrednim dostępem do danych, to w czasie jednego obiegu pętli programowej mielibyśmy na wyjściu dwie różne wartości. W przypadku sterowania nimi układu z pamięcią (przerzutnik, licznik) występowałyby błędy związane z niepożądanymi zbrociami sygnałów wyjściowych. Dlatego też program dla sterownika Intelster wyglądać może na przykład tak:

```

TH E 21 (testowanie wejścia RESET)
SJ 21 (ustawienie wyjścia Q)
RJ 22 (zerowanie wyjścia Q)
SPJ END (skok do końca, jeżeli wejście zerujące jest aktywne)
TH E 22 (testowanie wejścia SET)
SJ 22
RJ 21
END: SP 0

```

Jak widać, w tym programie występuje rozkaz skoku programowego, który powoduje omijanie testowania stanu wejścia ustawiającego (SET) w przypadku wykrycia jedynki na wejściu zerującym (RESET) - czyli wejściu o większym priorytecie.

Wracając jednak do sprzętowej realizacji dostępu do danych, można zastanowić się nad realizacją modułów we/wy, które łączyłyby, w jakiś sposób, dostęp bezpośredni i pośredni, a o wyborze trybu pracy całego modułu, czy też jego części, decydowałby użytkownik. Można by na przykład zrezygnować z rejestrów umieszczanych na kartach we/wy, a jednostkę centralną wyposażać w pamięć obrazu procesu i za pomocą odpowiedniego używania adresów korzystać albo z danych zawartych w tej pamięci (pośrednio), albo z danych wytworzonych na kartach (bezpośrednio). W ten sposób programista będzie mógł zdecydować o tym, jaki typ wejścia czy wyjścia użyje w danym przypadku. Można wtedy zachować możliwość wykonywania programu w sposób pseudorównoległy z równoczesnym obsługiwaniem niektórych (krytycznych) elementów systemu bezpośrednio.

4. NARZĘDZIA DO PISANIA PROGRAMÓW I OBSERWACJI PROCESU STEROWANIA

Istnieje wiele urządzeń, które umożliwiają użytkownikowi systemu PLC na "porozumiewanie" się ze sterownikiem lub grupą sterowników. Urządzenia te można ogólnie podzielić na trzy grupy, które jednak nie zawsze łatwo można wyodrębnić:

- programatory - urządzenia pozwalające na tworzenie programów sterowania, listy symboli, komentarzy, zbiorów dokumentacyjnych, przesyłanie danych i programów do sterownika, archiwizowanie ich na dyskach twardych i miękkich. Niektóre pozwalają testować program, który aktualnie realizowany jest przez sterownik, poprzez podgląd jego stanu wewnętrznego oraz możliwość zadawania wartości liczbowych przez programistę. Każda firma oferuje cały zestaw programatorów, które mają ułatwić pracę użytkownikowi. Są programatory przenośne, małe, o wymiarach zbliżonych do wymiarów kalkulatorów, jak na przykład PG605U, PG615U firmy Siemens czy P965 Firmy Modicon AEG. Inne mają kształt płaskich walizeczek z wyświetlaczem LC. Do tej grupy należą programatory PG 635 (Siemens), P125, P300, P600 (Modicon AEG). Są także produkowane duże, solidne programatory z monitorami ekranowymi i twardymi dyskami PG685, PG750, P500. Różne typy programatorów różnią się od siebie nie tylko wielkością i budową, ale także możliwościami funkcjonalnymi. I tak np. PG605U ze względu na małe rozmiary, mały wyświetlacz pozwala programować tylko w liście instrukcji (STL). Większe pozwalają na programowanie pamięci typu EPROM czy EEPROM;
- panele operatora (kontrolne) umożliwiające podglądanie programu, stanu procesu sterowania poprzez odczyt stanu rejestrów oraz zmiennych binarnych. Pozwolą one także na zadawanie zmiennych procesowych wprost z klawiatury operatora. Dla sterowników Modicon A można wymienić moduł operatora DBK021, czy też DBK111, natomiast dla sterowników Simatic S5 mały moduł OP393;
- panele wizualizacji umożliwiające podgląd stanu procesu w szacie graficznej wysokiej jakości. Dostępne są specjalne programy, które umożliwiają tworzenie modeli obiektów sterowania w postaci rysunków, na których można obserwować, jak zmienia się stan procesu aktualnie realizowanego. Firma Modicon AEG opracowała serię paneli wizualizacyjno - sterujących PanelMate Plus. Na zakończenie trzeba powiedzieć, że na rynku dostępne są także wersje firmowego oprogramowania przeznaczonego dla komputerowego wspomaganie pisania programów, kontroli i wizualizacji. Tak więc komputer klasy IBM może być wykorzystywany zarówno jako programator, panel operatora, urządzenie kontrolujące lub wizualizujące.

5. PODSUMOWANIE

Możliwości, jakie dają poszczególne programy przeznaczone do pracy ze sterownikami programowalnymi oraz sposoby dostępu do danych są różne, ale widać, że producenci dają dość dużą swobodę co do formy przedstawiania programów (ich struktury). Należy te możliwości jak najlepiej wykorzystać w celu stworzenia optymalnego oprogramowania procesowego. Od strony sprzętowej natomiast pomaga programiście, czy operatorowi systemu, szeroka gama urządzeń programujących, testujących i wizualizujących. Ostatnio producenci sterowników programowalnych zaczynają myśleć o stworzeniu systemu otwartego, który znormalizowałby wszystko, co dotyczy sprzętu oraz oprogramowania sterowników [7,8], jednak nie wszystkie firmy są jeszcze tym zainteresowane. Takie znormalizowanie na pewno pomogłoby programistom w pracy ze sterownikami.

Wpłynęło do Redakcji w czerwcu 1994 r.

LITERATURA

- [1] Intelster PC 4k. Opis techniczny systemu. Mera ZAP, Ostrów Wielkopolski.
- [2] 984 - A120 Compact Programmable Controllers, User's Guide.
- [3] Modicon 984 Programmable Controller, Systems Manual.
- [4] Programmable Controller Simatic S5 - 100U, Manual.
- [5] PG 675 Programmer Simatic S5, Manual.
- [6] Berger H.: Automatykacja za pomocą sterownika SIMATIC S5-115U.
- [7] Blickley G.J.: New developments in programmable controllers and peripherals. Control Engineering, 1987, January.
- [8] Quatse J. T.: Programmable Controllers of the future. Control Engineering, 1986, January.

Abstract

The article presents certain problems related to creating programs for programmable controllers. It describes methods of partitioning of a program into smaller parts, employed in the STEP-5 and Modsoft packages, used for programming Siemens and Modicon-AEG programmable controllers. Next, the problem of access to process data and its influence on the "shape" of the program is discussed. The advantages and disadvantages of direct and indirect access, and the possibility of constructing I/O cards with selection of the method of data read and write are presented. At the end a short review of hardware tools for program development and visualization is given. Special attention is paid to programmers, operator and visualization systems.