

*mgr inż. Magdalena PORADA*  
*Akademia Muzyczna im. F. Chopina*  
*Warszawa*

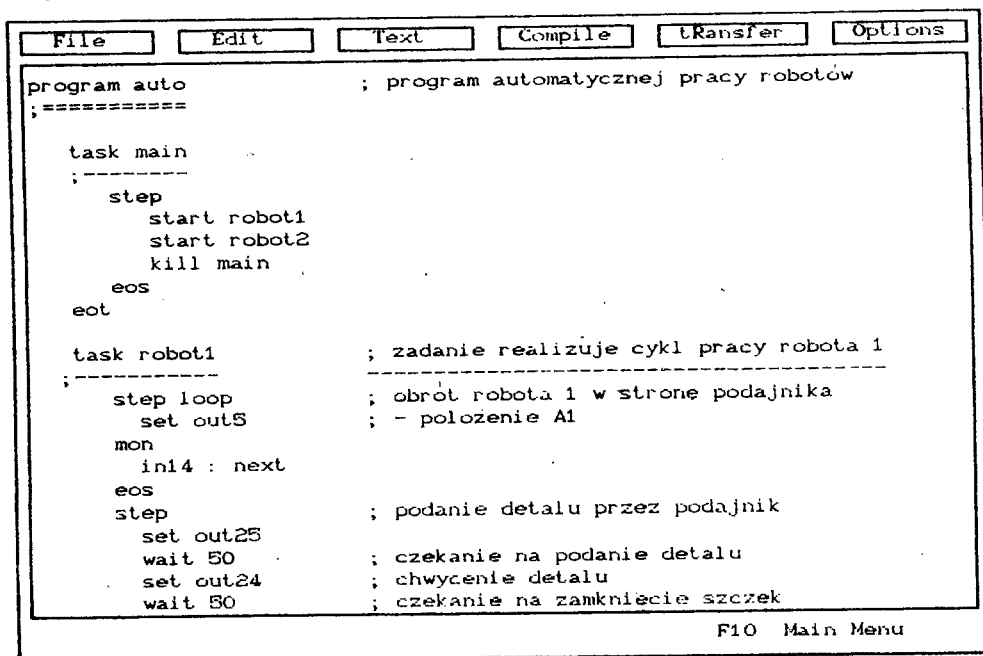
*mgr inż. Marek ALEKSANDRUK*  
*mgr inż. Lesław KOŁAKOWSKI*  
*Przemysłowy Instytut Automatyki*  
*i Pomiarów MERA-PIAP*  
*Warszawa*

## SYSTEM KOMPILACJI JĘZYKA EARL

### 1. WSTĘP

System kompilacji umożliwia przygotowanie programu użytkowego dla sterownika F-PLC w prosty i wygodny sposób. System ten realizuje:

- edycję i kompilację programów użytkowych napisanych w języku EARL na postać wynikową (ładowaną do pamięci sterownika),
- ładowanie kodu wynikowego do pamięci sterownika poprzez łącze szeregowe RS232C.



```
File Edit Text Compile Transfer Options

program auto ; program automatycznej pracy robotów
;=====

task main
;-----
step
start robot1
start robot2
kill main
eos
eot

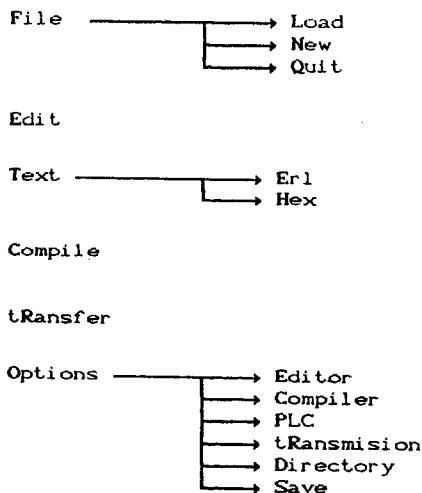
task robot1 ; zadanie realizuje cykl pracy robota 1
;-----
step loop ; obrót robota 1 w stronę podajnika
set out5 ; - położenie A1
mon
in14 : next
eos
step ; podanie detalu przez podajnik
set out25
wait 50 ; oczekiwanie na podanie detalu
set out24 ; chwycenie detalu
wait 50 ; oczekiwanie na zaniknięcie szczek

F10 Main Menu
```

Rys. 1. Ekran konwersacji z użytkownikiem

## 2. WARSTWA KONWERSACJI Z UŻYTKOWNIKIEM

System kompilacji stanowi osobny pakiet programowy, przeznaczony do pracy na mikrokomputerach kompatybilnych z IBM PC. Wywołany jest przez podanie komendy EARL. W górnej części ekranu pojawia się menu (rys. 1) – pozostała przestrzeń przeznaczona jest na okno, które może zawierać tekst programu. Użytkownik wydaje polecenia przez wybranie odpowiedniej pozycji w menu systemu. Aktualnie wybrane polecenie jest podświetlone. Naciśnięcie klawisza *Enter* powoduje jego realizację. Wybranie polecenia w menu odbywa się poprzez ustawienie kursora w pozycji odpowiadającej temu poleceniu albo przez naciśnięcie klawisza z wyróżnioną literą w nazwie polecenia. Klawisz *Esc* służy do opuszczania danego poziomu menu. Struktura menu systemu pokazana jest na rys. 2. Pracę z systemem w każdej chwili można przerwać przez wydanie polecenia *Quit* (podfunkcja pola *File* w menu systemu).



Rys. 2. Struktura menu systemu

## 3. PRACA Z SYSTEMEM KOMPILACJI

### 3.1. Początek pracy z systemem

Po wywołaniu systemu użytkownik musi podać nazwę pliku z jakim zamierza pracować. W tym celu należy wykonać polecenie *LOAD* (jako podfunkcję pola *FILE* w menu systemu). W dolnej części ekranu pojawi się okno z pytaniem o nazwę pliku. W oknie tym należy wpisać nazwę pliku. Należy wprowadzić nazwę bez rozszerzenia. System przyjmie nazwę pliku z domyślnym rozszerzeniem *ERL*. Po wpisaniu nazwy i naciśnięciu klawisza *Enter* na ekranie pojawi się tekst programu zapisany w podanym zbiorze. Jeżeli zbiór o podanej nazwie nie istnieje, wówczas system zasygnalizuje błąd i będzie oczekiwał na nowe polecenie. W przypadku, gdy dopiero zamierza się przystąpić do pisania tekstu programu od początku (nie istnieje plik z tekstem programu),

należy wykonać polecenie NEW. Jest to również funkcja w obrębie opcji FILE. Po wydaniu tego polecenia system zapyta o nazwę pliku (podobnie jak w poleceniu LOAD). Tutaj również należy podać nazwę bez rozszerzenia i przyjęte zostanie rozszerzenie domyślne ERL.

### 3.2. Edycja tekstu programu

Edycję tekstu programu można prowadzić po wydaniu polecenia EDIT. System kompilacji nie dysponuje własnym edytorem znakowym, natomiast pozwala na wywołanie „z wnętrza” systemu dowolnego edytora, jakim dysponuje i do jakiego jest przyzwyczajony użytkownik. Zainstalowanie wybranego edytora w systemie odbywa się przez wykonanie polecenia Edit w ramach opcji Options w menu systemu.

Jeżeli użytkownik nie zamierza zmieniać tekstu programu, natomiast zależy mu na jego przeglądaniu, może posłużyć się poleceniem Text. Do przeglądania tekstu służą klawisze  $\uparrow$   $\downarrow$  *Page Up*, *Page Down* oraz *Home* (przejdzie do początku tekstu) i *End* (przejdzie do końca). Opcja Text w menu systemu pozwala na przełączenie postaci oglądanego programu z tekstu źródłowego (ERL) na tekst kodu wynikowego w postaci heksadecymalnej (HEX).

### 3.3. Kompilacja programu

Zadaniem kompilatora jest sprawdzenie poprawności programu źródłowego i jego przełożenie na postać wynikową, nadającą się do wykonania przez sterownik. Program źródłowy musi być napisany w języku EARL.

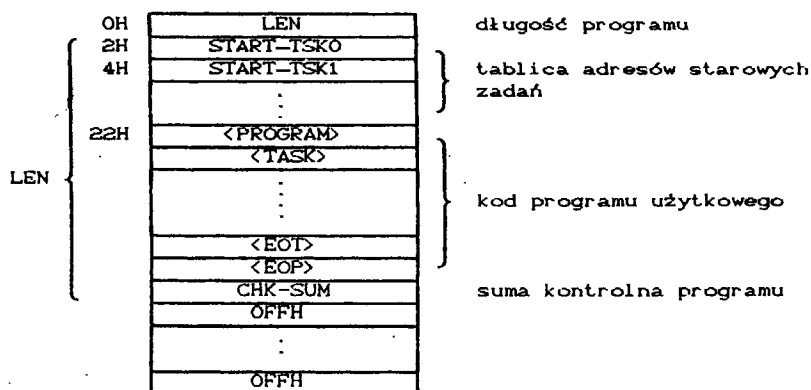
Kompilacji programu dokonuje się poleceniem Compile w menu systemu. Jeżeli kompilator znalazł w programie błąd, wtedy na ekranie pojawia się okno zawierające informację o rodzaju błędu z podaniem numeru linii w programie źródłowym, w której wykryto błąd. W takim przypadku użytkownik musi przejść do edytora i poprawić tekst programu. Poprawienie jednego błędu nie gwarantuje jeszcze, że program jest poprawny, ponieważ kompilator przerywa analizę programu po wykryciu pierwszego błędu. Jeżeli program nie zawiera żadnego błędu, wtedy na ekranie pojawi się informacja, że program jest poprawny.

W wyniku kompilacji poprawnego programu w katalogu użytkownika zostaną założone pliki zawierające kod wynikowy, o nazwie takiej, jak plik źródłowy i o rozszerzeniach BIN i HEX. Plik wyjściowy typu HEX zawiera program w kodzie wykonywalnym, zapisany w postaci rekordów w formacie INTEL—HEX. Każdy rekord programu jest opatrzony komentarzem, który zawiera nazwę symboliczną instrukcji i typ parametrów tej instrukcji. Plik wyjściowy typu BIN zawiera program w kodzie wykonywalnym zapisany w postaci obrazu pamięci.

Użytkownik, posługując się poleceniem Options, może wybrać, który z plików wyjściowych ma zostać utworzony oraz może zadeklarować wielkość układu pamięci,

w którym ma zostać zapisany program. Jeżeli program wynikowy nie zmieści się w zadeklarowanej pamięci, wówczas system zasygnalizuje, że program jest zbyt duży. W takim wypadku należy sprawdzić, czy rzeczywiście zadeklarowany rozmiar pamięci odpowiada istniejącej w sterowniku. Jeżeli tak, to należy zwiększyć pamięć sterownika lub zoptymalizować program.

Kod ładowalny w postaci binarnej (rozszerzenie BIN) jest bajt po bajcie obrazem pamięci programu użytkowego sterownika. Rozmiar zbioru jest taki, jak zadeklarowana przed konwersją wielkość układu pamięci. Zbiór ten zawiera kod interpretowanego programu, tablicę adresów startowych poszczególnych zadań, oraz dane kontrolne programu takie jak długość i suma kontrolna. Niewykorzystanym bajtom pamięci jest nadawana wartość 0FFH. Rys. 3 przedstawia strukturę zbioru typu BIN. Zbiór typu BIN może być bezpośrednio zapisany do pamięci EPROM za pomocą dowolnego programatora takich pamięci.



Rys. 3. Struktura zbioru typu BIN

Drugi zbiór wyjściowy kompilatora z kodem ładowalnym (o domyślnym rozszerzeniu HEX) jest zbiorem rekordów tekstowych o formacie INTEL-HEX. Rekordy składają się ze znaków kodu szesnastkowego. Każdy bajt jest kodowany na dwóch znakach. Ogólna postać rekordu jest następująca:

:<BYTE NUM> <ADR> <TYPE> <byte> <byte>..<byte> <CHK SUM>

BYTE NUM – liczba bajtów zakodowanych w polu danych (2 znaki),

ADR – adres ładowania pierwszego bajtu danych (4 znaki),

TYPE – typ rekordu (2 znaki),

byte – bajt danych (2 znaki),

CHK SUM – suma kontrolna wszystkich bajtów rekordu (bez uwzględnienia znaku ":") (2 znaki).

Kompilator tworzy zbiór, który składa się z rekordów danych (TYPE = 00) i jednego, umieszczonego na końcu, rekordu końca danych (TYPE=01). Ponieważ liczba bajtów

danych każdego rekordu jest zapisana w tym rekordzie, to wszystkie dodatkowe znaki zapisane przed znakiem końca linii są pomijane i mogą być traktowane jako komentarz. W zbiorze typu HEX tworzonym przez kompilator komentarzem jest kod rozkazu w postaci symbolicznej wraz z typem występujących po nim parametrów (rys. 4).

```

: 20400200FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFBE.....REKORD INICJALIZACJI TABLICY ZADAN
: 0240020022409E.....REKORD TABLICY OPISU ZADAN, zadanie nr 0
: 02402200780024.....<TASK ><nr8>
: 0140230002DA.....<STEP ><void>
: 024023000D19B4.....<SETO ><nr8>
: 0140260003D6.....<MON ><void>
: 0240270021199D.....<PUSHI><nr8>
: 0340290008402D5F.....<GIT ><adr16>
: 01402C0004CF.....<EOS ><void>
: 01402D0002D0.....<STEP ><void>
: 02402E001119A6.....<RESO ><nr8>
: 0140300003CC.....<MON ><void>
: 024021002D1987.....<NOTI ><nr8>
: 034033000840235F.....<GIT ><adr16>
: 01403B0004C5.....<EOS ><void>
: 0140370005C3.....<EOT ><void>
: 02403800003650.....REKORD DLUGOSCI PROGRAMU
: 0140B4003A01.....REKORD SUMY KONTROLNEJ PROGRAMU
: 00000001FF.....REKORD KONCA ZBIORU

```

*Rys. 4. Przykład zbioru typu HEX*

Zbiór typu HEX może być przesyłany łączem szeregowym RS232C z komputera do sterownika, gdzie odpowiedni fragment oprogramowania systemowego przetwarza go i zapisuje do pamięci EEPROM. Transmisję realizuje funkcja tRansmit w menu systemu.

### 3.4. Transmisja kodu wynikowego do pamięci sterownika

Wykonanie polecenia tRansmit powoduje przesłanie łączem szeregowym RS232C zawartości zbioru typu HEX do pamięci sterownika. Parametry transmisji ustawiane są przez wykonanie polecenia tRansmisja w polu Options w menu systemu. Po wykonaniu polecenia system wypisuje komunikat informujący o pomyślnym przebiegu transmisji lub też wskazujący na przyczynę, która prawidłową transmisję uniemożliwiła:

- nie został znaleziony zbiór z kodem wynikowym w postaci heksadecymalnej,
- źle zostały ustawione parametry transmisji,
- nie jest aktywne łącze szeregowe,
- został przepełniony bufor po stronie odbiorcy,
- w czasie transmisji wystąpiły błędy.

#### 4. KONFIGURACJA SYSTEMU

System konfiguruje się (dostosowuje do konkretnego przypadku) przez wykonywanie różnych funkcji w ramach pola Options w menu systemu.

- Editor  
Ta opcja pozwala na określenie edytora, jakim użytkownik zamierza się posługiwać. W tym celu należy podać nazwę edytora (wraz z pełną ścieżką dostępu i rozszerzeniem).
- Compiler  
Wybranie tej opcji pozwala określić, jakie pliki wynikowe mają być tworzone w wyniku kompilacji programu (HEX, BIN, oba).
- PLC  
Ta opcja pozwala na określenie wielkości pamięci sterownika.
- tRansmission  
Przy użyciu tej opcji użytkownik może określić:
  - sposób przesyłania łączy szeregowych kodu programu użytkownika do sterownika (Method). Program może być przesyłany metodą bezpośrednią (Direct) lub pośrednią (Indirect). W metodzie bezpośredniej komputer połączony jest łączy szeregowym RS232C wprost z modułem, do którego ładowany jest program. W metodzie pośredniej komputer połączony jest łączy szeregowym z modułem komunikacyjnym. Moduł ten retransmituje po magistrali ISB otrzymywany kod do odpowiedniego innego modułu sterownika. Metoda pośrednia wymaga podania numeru modułu docelowego, co jest możliwe w ramach opcji tRansmission. Metoda pośrednia ma zastosowanie tylko wtedy, gdy sterownik jest wyposażony w moduł komunikacyjny. W przeciwnym przypadku należy posłużyć się metodą bezpośrednią.
  - prędkość transmisji. Prędkość transmisji w metodzie bezpośredniej określamy doświadczalnie. Moduł sterownika F-PLC ustawia maksymalną standardową dopuszczalną, ze względu na rodzaj użytej pamięci programu, prędkość transmisji. Jeżeli przesyła się kod z niewłaściwą prędkością, wtedy sterownik sygnalizuje błąd i należy ustawić inną prędkość.
- Directory  
Opcja ta umożliwi określenie położenia katalogu użytkownika (User), w którym znajdują się jego projekty, oraz katalogu systemowego (System), gdzie są zapisane elementy systemu kompilacji.
- Save  
Wykonanie tego polecenia powoduje, że wprowadzona konfiguracja będzie obowiązywać przy ponownych uruchomieniach systemu.
- Options  
Wyświetla aktualny stan wszystkich opcji.