



graficzny kompilator dla '51

Nasze artykuły o ST6-Realizerze wywołały spore zainteresowanie wśród Czytelników EP. Okazuje się jednak, że nie trzeba sięgać aż do Holandii i "zmuszać się" do mikrokontrolerów ST62! Program realizero-podobny dla '51 napisał Polak i - co bardzo ważne - rozdaje go za darmo! W artykule prezentujemy uwagi doświadczonego projektanta na temat WireIt!

Proces programowania jest coraz częściej wspomagany i ułatwiany przez narzędzia pracujące w środowisku graficznym i automatyzujące wiele podstawowych funkcji i czynności. Tak jest zarówno na poziomie „dużych” komputerów (Visual C, Visual Basic, Delphi) jak i na szczeblu niewielkich mikrokontrolerów jednocukłowych (np. ST Realizer). Nowy program „WireIt!” - przeznaczony na razie dla serii '51 - zakłada maksymalną eliminację kodu „pisanego” na rzecz operowania gotowymi blokami logicznymi, łączonymi graficznie w większe struktury zgodnie z potrzebami naszej aplikacji. Metoda taka (znana np. ze sterowników LOGO! lub pakietu Genie) umożliwia stworzenie programu bez głębszej znajomości struktury mikrokontrolera i jego listy rozkazów.

Obsługa jest prosta i wymaga jedynie przeczytania kilku akapitów w pomocy. Po uruchomieniu mamy przed sobą pole projektowe bloku z paskiem wejść na górze i paskiem wyjść na dole. Autor dla wejść/wyjść wprowadza wspólną nazwę „port” (której nie należy mylić z konstrukcyjnymi portami mikrokontrolera). Może ona dotyczyć zarówno kolejności wykonywania instrukcji (port typu thread - wątek), jak też przekazywania zmiennych (porty typu 8-bit, 16-bit, boolean - logiczne). Poszczególne typy portów są wyróżnione kolorami. Podstawową minimalną konfiguracją jest jedno wejście i jedno wyjście typu wątek (jak na rys. 1) - blok nie otrzymuje i nie wyprowadza żadnych wartości. Jest to kompletna struktura całego, chociaż bardzo nie-

Jak widać jest to zgodne z graficznym przebiegiem wątku: deklaracja stałej 1 (adres 126), deklaracja stałej 2 (adres 127), wykonanie dodawania (wynik pod adresem 125) i zatrzymanie programu w pętli (blok Sink 1).

Sam proces projektowania polega na umieszczeniu niezbędnych komponentów na ekranie i połączeniu odpowiednich portów wyjściowych z wejściowymi metodą „przeciągnij i upuść”.

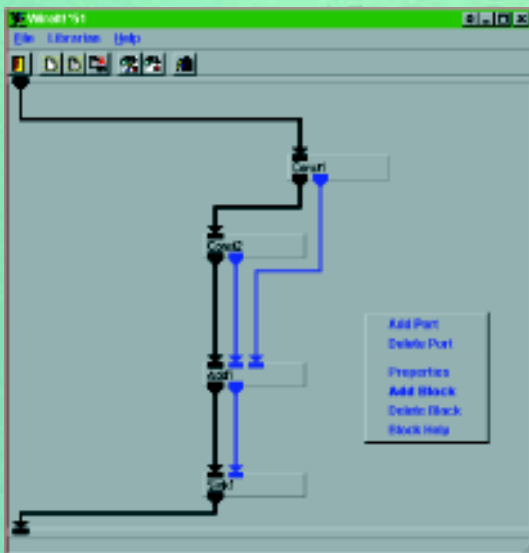
Manipulację komponentami umożliwia menu kontekstowe również widoczne na rys. 1. Po wybraniu *Add Port* otwiera się okno dialogowe wyświetlaczy alfanumerycznych LCD, a także zbiór procedur modułu edukacyjnego DSM-51. Jest to rzecz jasna zbyt mało, nawet jak na potrzeby nieskomplikowanego programu, poza tym w polu projektowym nie zmieści się zbyt wiele bloków. Dlatego też z elementów podstawowych (portów i bloków) składamy większe potrzebne nam struktury, które kompilujemy do postaci własnego bloku (pliki z rozszerzeniem *.wio) i dołączamy do biblioteki. Potem zaś możemy już korzystać z nowej struktury tak samo, jak z bloków gotowych. Należy dodać, że przewidziane są także bloki pozwalające na wpisanie własnych fragmentów kodu assemblerowego. Może to znacznie usprawnić pracę (choć jest nieco sprzeczne z podstawowymi założeniami programu).

Do wyboru mamy cały szereg predefiniowanych bloków. Oprócz podstawowych funkcji arytmetycznych i struktur logicznych (jak np. *if*) do dyspozycji są procedury obsługi wyświetlaczy alfanumerycznych LCD, a także zbiór procedur modułu edukacyjnego DSM-51. Jest to rzecz jasna zbyt mało, nawet jak na potrzeby nieskomplikowanego programu, poza tym w polu projektowym nie zmieści się zbyt wiele bloków. Dlatego też z elementów podstawowych (portów i bloków) składamy większe potrzebne nam struktury, które kompilujemy do postaci własnego bloku (pliki z rozszerzeniem *.wio) i dołączamy do biblioteki. Potem zaś możemy już korzystać z nowej struktury tak samo, jak z bloków gotowych. Należy dodać, że przewidziane są także bloki pozwalające na wpisanie własnych fragmentów kodu assemblerowego. Może to znacznie usprawnić pracę (choć jest nieco sprzeczne z podstawowymi założeniami programu).

Na wygenerowaniu pliku *.asm wspomaganie programowania niestety się kończy. Dalsze etapy (kompilacja do postaci Intel Hex, ewentualne symulacje, ładowanie do pamięci) muszą być wykonane za pomocą tradycyjnych narzędzi. Być może w przyszłych wersjach pojawią się dodatkowe rozszerzenia.

Dla osób dopiero zapoznających się z programowaniem mikrokontrolerów *WireIt!* będzie sporą pomocą. Pozwoli na zrozumienie logiki „postępowania” procesora, porównanie wykonywanych operacji z generowanym kodem, uruchomienie pierwszych własnych aplikacji. Nie jest natomiast ofertą dla elektroników już zaznajomionych z rodziną MCS-51.

Własne oprogramowanie w assemblerze lub C daje bowiem nieporównanie większą elastyczność i możliwości. Kod generowany automatycznie ma - pomimo całego uznania dla twórcy programu - wiele ograniczeń, które będą dokuczliwe dla bardziej doświadczonych. Np. zmienne logiczne (boolean) są reprezentowane bajtowo - nie wykorzystują silnej strony '51, jaką są operacje na pojedyn-



Rys. 1.

wielkiego, programu.

Wygenerowany dla tego programiku kod jest następujący:

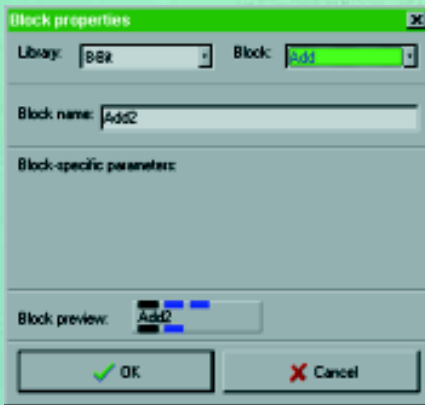
```

; Designed and compiled using WireIt!
'51
      AJMP   ProgramEntry
      ORG    100H
; Compilation ID: 97336174
; Const1 of type 8-Bit.Const
ProgramEntry:
      MOV    126, #0
; Const2 of type 8-Bit.Const
Const2_97336174_stdin:
      MOV    127, #16
; Add1 of type 8-Bit.Add
Add1_97336174_stdin:
      MOV    A, 127
      ADD    A, 126
      MOV    125, A
; Sink1 of type 8-Bit.Sink
Sink1_97336174_stdin:
      AJMP   ProgramExit
ProgramExit:
      SJMP   ProgramExit

```



Rys. 2.



Rys. 3.

czych bitach (flagach). Nie bardzo też wiem, jak zgrabnie umieścić w graficznym „strumieniu“ bloków obsługę przerw, bez których skuteczność większości aplikacji jest istotnie zmniejszona. Sam mechanizm graficzny też może być uciążliwy dla przyzwyczajonych do kodu tekstowego: np. zapis prostej operacji *if (k) a=b+c; else a=b-c* zajmie pół ekranu. Również poruszanie się wśród większej liczby ekranów (bloków) raczej nie będzie prostsze od korzystania z procedur i makroinstrukcji.

Oprócz tych uwag podstawowych nasuwa się także kilka spostrzeżeń dotyczących bezpośrednio interfejsu oraz wygody obsługi. Zaznaczam, że mają one charakter subiektywny. Poza tym program jest na razie prezentowany w wersji *beta* (czyli na etapie rozwoju i wprowadzania zmian i poprawek). Komponowanie graficzne projektu byłoby łatwiejsze gdyby: po pierwsze - uwidocznić przeciąganie bloków (np. tak jak to jest zrealizowane w Delphi), zaś po drugie - zaznaczyć w jakiś sposób (zmianą koloru, miganiem) port wyjściowy podlegający właśnie łączeniu.

Wydaje się, że te udogodnienia nie powinny być zbyt kłopotliwe od strony programowej. Warto chyba też rozważyć przyspieszenie procesu wstawiania nowych bloków. Za każdym razem bowiem otwiera się okienko dialogowe w podstawowym ustawieniu. Wybranie opcji dla kilku kolejnych jednakowych bloków np. typu *const* robi się cokolwiek monotonne. Może dla typowych bloków wprowadzić dodatkowe paski narzędziowe? Dobra byłaby też funkcja kopiowania bloków już umieszczonych na pulpicie.

Dla mnie osobiście trochę zbyt mało czytelne są pliki pomocy. Sporządzone w formacie HTML posiadają atrakcyjną szatę gra-

ficzną. Nadaje się ona znakomicie do prezentacji w witrynie WWW, natomiast - niestety - nieco mniej do szybkiego korzystania w trakcie pracy. Pod tym względem klasyczna windowsowa pomoc byłaby skuteczniejsza.

I na koniec generalna uwaga: oferując program polskiemu odbiorcy trzeba go będzie wyposażyć w odpowiednią wersję językową. Poszerzy to niewątpliwie grono użytkowników, będzie też zgodne z panującymi trendami rynkowymi i prawnymi.

Niech te skrótowe uwagi zostaną potraktowane nie jako krytyka produktu, ale jako sugestie użytkowników - zresztą zbieranie opinii Autor przewidział i wsparł opcją bezpośredniego wysyłania poczty na swój adres.

Program *WireIt!* autorstwa Stanisława Skowronka jest w testowej wersji beta dostępny na stronie www.ep.com.pl. Znajdzie się także na płycie CD-EP8.

Jerzy Szczesiul, AVT
jerzy.szczesiul@ep.com.pl

Program WireIt! jest dostępny w Internecie, pod adresami friko4.onet.pl/po/thesis/wireit.html, www.ep.com.pl/ftp/wireit.exe oraz na płycie CD-EP8.