

Realizacja projektów na 8051 przy pomocy oprogramowania firmy

W tej części cyklu prezentujemy, jak w prosty sposób można utworzyć program przy pomocy systemu IAR Embedded Workbench, pracujący pod nadzorem Windows.

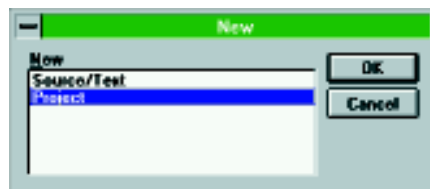
Pierwszy program....

Wszystkie zbiory: źródłowe, obiektowe, konfiguracyjne oraz wynikowe są zorganizowane w większą grupę zwaną projektem. W systemie shella Embedded Workbench taka architektura przy tworzeniu projektu użytkownika jest najbardziej optymalna.

Pierwszym zatem krokiem do utworzenia nowego programu (źródła) jest stworzenie w systemie IAR projektu. Następnie należy zdefiniować typ procesora, na który dedykowany jest tworzony przez nas program. Wreszcie należy utworzyć listę plików wejściowych projektu (źródłowych). Ta ostatnia będzie w większości przypadków tworzona na etapie pisania kodu źródłowego programu.

Utworzenie nowego projektu

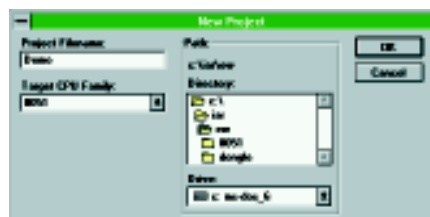
Aby utworzyć nowy projekt należy wybrać opcję **New...** z menu **File**, na ekranie pojawi się okno - patrz **rys.1**.



Rys. 1.

Wybieramy opcję **Project**, a następnie potwierdzamy wybór klawiszem **OK**.

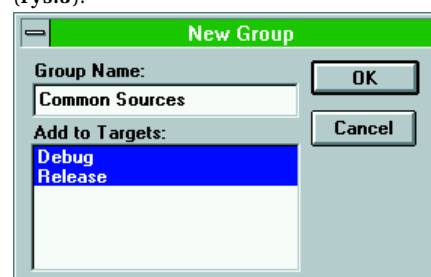
Nowy projekt dedykowany będzie na procesor z rodziny 8051. Ustawienia tego parametru dokonujemy w oknie **New Project** modyfikując opcje: **Project Filename** oraz **Target CPU Family** (**rys.2**).



Rys. 2.

Klawiszem **OK** potwierdzamy wybór, projekt zostaje utworzony. Jego struktura po każdej aktualizacji jest wyświetlana w oknie **DEMO.PRJ**, które jest dostępne w systemie w każdej chwili.

Następnym krokiem jest utworzenie grupy zawierającej listę zbiorów źródłowych (ich nazw), które potem utworzymy używając wbudowanego edytora (**rys.3**).



Rys. 3.

Wprowadzanie nazw zbiorów źródłowych odbywa się w opcji **Files...** z menu **Project**. W naszym przypadku zbiory demonstracyjne zostały wcześniej utworzone, będą to dwa zbiory: **demo.c** i **demo_two.c**.

Dołączenia tych zbiorów do listy dokonujemy myszką oraz potwierdzamy klawiszem **Add**. Po wybraniu wszystkich zbiorów (w naszym przypadku dwóch) kończymy operację naciskając **Done** (**rys.4**).



Rys. 4.

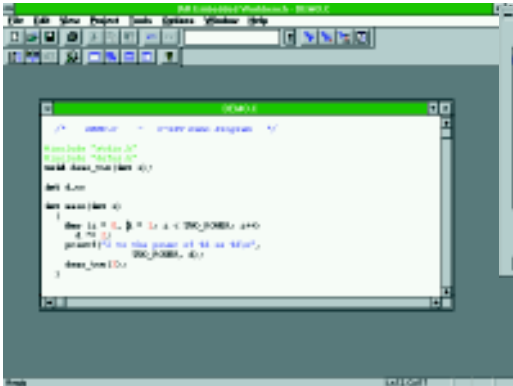
W oknie projektu pojawi się struktura naszego projektu. Oczywiście w dowolnej chwili można tę strukturę modyfikować dodając nowe elementy. Postępujemy wtedy w taki samo sposób korzystając z opcji **Files** z menu **Project** (**rys.5**).



Rys. 5.

Edycja zbiorów źródłowych

Aby wyedytować dowolny zbiór z naszego projektu, to po prostu „klikamy“ myszką na jego nazwę w oknie projektu. **Rys.6** pokazuje okno edycji zbioru **DEMO.C**.



Rys. 6.

Edycja programu w systemie Embedded Workbench jest szczególnie przyjemna. Wszystkie słowa specjalne są wyróżnione odmiennym kolorem, co ułatwia wzrokową analizę kodu programu. I tak:

- słowa kluczowe są pogrubione;
- tekst komentarza wyświetlony jest w kolorze ciemnym niebieskim;
- dyrektywy preprocesora wyróżnione są na zielono;
- stałe w równaniach i wyrażeniach numerycznych - na czerwono;
- stringi - kolor niebieski;
- pozostałe elementy programu są w kolorze czarnym.

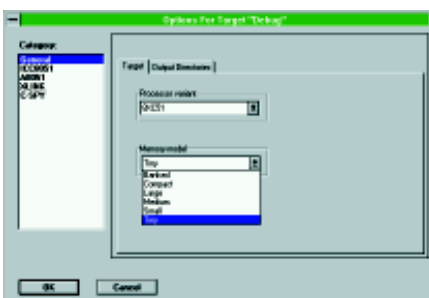
Po utworzeniu programu źródłowego można przystąpić do jego kompilacji.

Kompilowanie projektu

System Embedded Workbench pozwala na ustalenie opcji dla całego projektu, wybranej grupy zbiorów lub dla każdego zbioru osobno. My w naszym przykładzie ustalimy parametry kompilacji dla całego projektu **Release**, dzięki czemu nie będzie konieczne ustawianie ich osobno dla każdego elementu.

W oknie projektu należy wybrać folder **Release**, a następnie otwieramy opcję **Options...** z menu **Project**.

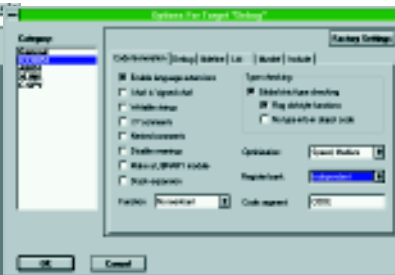
Wyświetlone zostaje okno opcji, z którego wybieramy podopcję **General** w liście **Category** (**rys.7**).



Rys. 7.

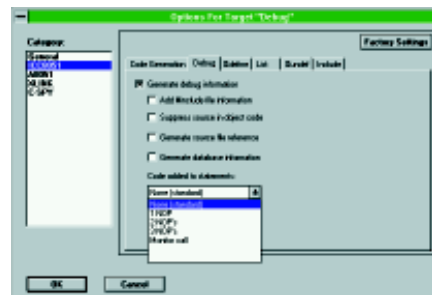
Dla naszego projektu wybieramy wariant procesora **8XC51** oraz najmniejszy model pamięci **Tiny**.

Następnie otwieramy kategorię kompilatora **ICC8051** w celu ustawienia opcji dla tego programu (**rys.8**).



Rys. 8.

Ponieważ w kolejnym odcinku przedstawimy pracę z debuggerem **C-SPY**, włączamy opcję „**Generate debug information**“. Powoduje to wygenerowanie zbioru z informacjami niezbędnymi do przeprowadzenia analizy naszego programu z debuggerem (**rys.9**). Potwierdzamy wybór naciskając klawisz **OK**.



Rys. 9.

Teraz możemy skompilować wszystkie programy źródłowe, wybierając za każdym razem odpowiedni moduł **.C** w oknie projektu. Skompilowanie następuje przez wybór opcji **Compile** z menu **Project** lub poprzez naciśnięcie kombinacji klawiszy **Ctrl-F9**. Wynik kompilacji przedstawiony zostaje w dodatkowym okienku **Messages** (**rys.10**).

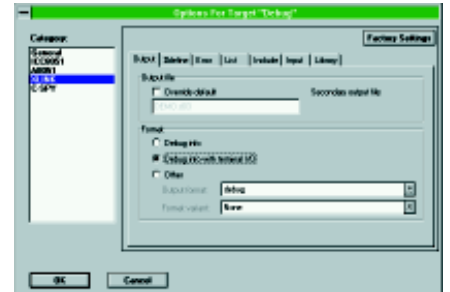


Rys. 10.

Ponieważ program źródłowy napisany był bezbłędnie, komunikat wskazuje na poprawne zakończenie kompilacji.

Po skompilowaniu wszystkich plików składowych naszego projektu można przystąpić do ich połączenia - **LINKING**.

Podobnie jak w przypadku kompilatora, przed tą operacją należy ustawić parametry konsolidacji używając opcji **Options** z menu **Project**. Wybieramy kategorię **XLINK**, aby wyświetlić w oknie dostępne opcje linkera (**rys.11**).



Rys. 11.

Zaznaczamy myszką opcję „**Debug info with terminal I/O**“, ze względu na późniejsze użycie debuggera **C-SPY**. Potwierdzamy wybór wskazując myszką **OK**.

Aby wykonać linkowanie wybieramy opcję **Link** z menu **Project**. Wygenerowany zostaje zbiór **aout.d03** który potem użyjemy do pracy z debuggerem **C-SPY**. Poprawność wykonania operacji można sprawdzić w oknie **Messages** podobnie jak przy kompilacji kodu źródłowego (**rys.12**).



Rys. 12.

W ten prosty sposób utworzyliśmy nasz pierwszy, dość prosty projekt. W następnej części przedstawimy sposób posługiwania się programem **C-SPY**.

Sławomir Surowiński, AVT