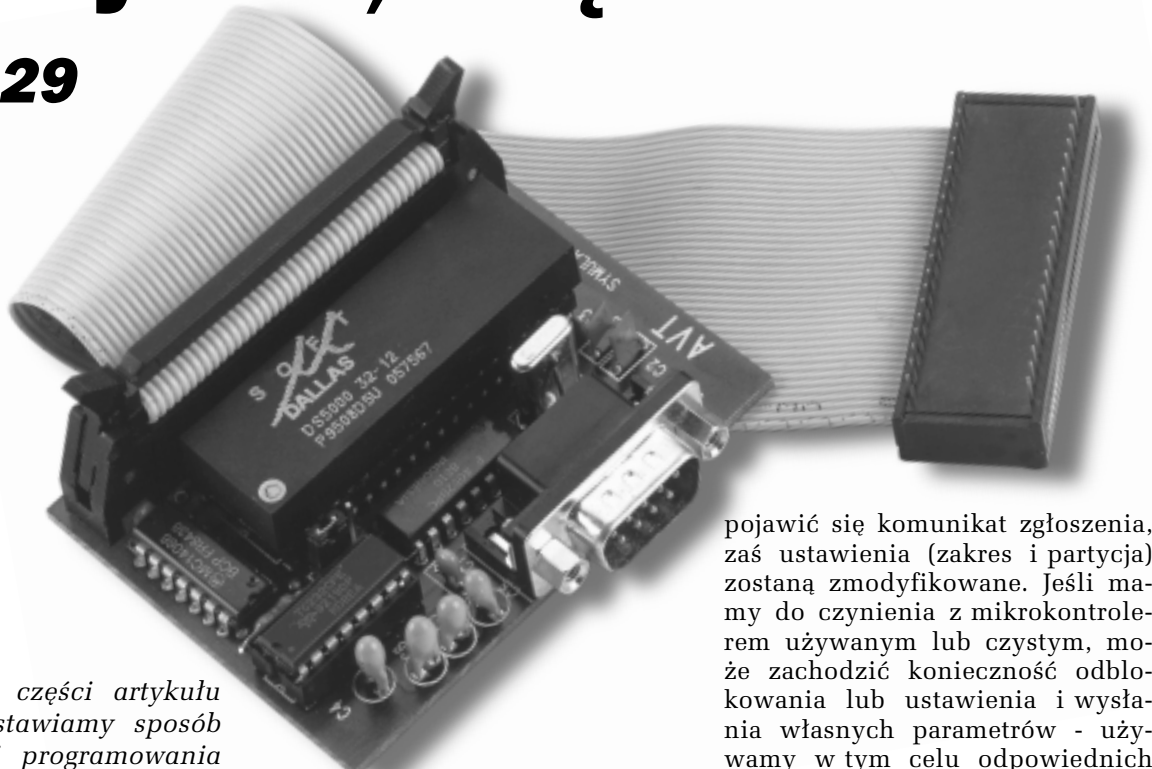


Emulator DS5000 i innych mikrokontrolerów rodziny '51, część 2

AVT-5029



W drugiej części artykułu przedstawiamy sposób montażu i programowania emulatora wykonanego w oparciu o niezwykle mikrokontroler - DS5000.

Uruchomienie zestawu

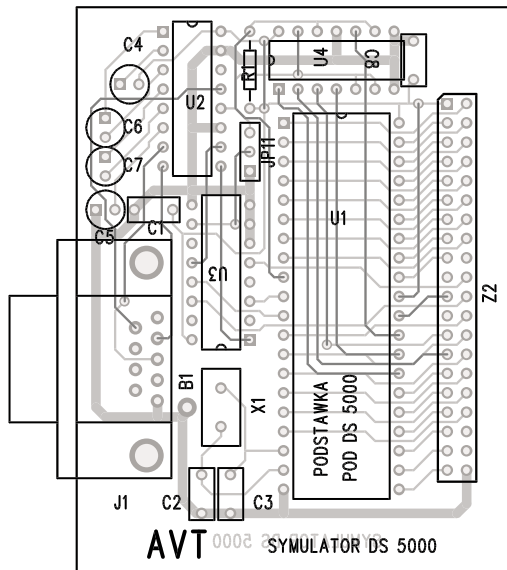
Po złożeniu i wstępnym sprawdzeniu płytki (schemat montażowy na rys. 6) podłączamy ją do wolnego portu szeregowego komputera kablem null-modem (potrzebne są tylko linie GND, TxD, RxD, DTR). Wtyk testowy wkładamy do dowolnej płytki testowej (uwaga na kierunek!). Do prób może to być zwykła podstawka dyl z podłączonym zasilaniem +5V i jakimś kontrolnym LED-em do migotania. Włączamy zasilanie. W oknie konfiguracji loadera ustawiamy odpowiedni port oraz szybkość (domyślna wynosi 19200 baud, możemy próbować na 57600, ale nie zawsze udaje się bez zrywania).

Wybieramy przygotowany uprzednio dowolnym narzędziem plik Intel Hex (po kliknięciu na pasku opisu pliku otwiera się okno wyboru z domyślnymi rozszerzeniami *.hex oraz *.ihx). Następnie klikamy „Pobierz status“ - w okienku podglądu powinien

pojawić się komunikat zgłoszenia, zaś ustawienia (zakres i partycja) zostaną zmodyfikowane. Jeśli mamy do czynienia z mikrokontrolerem używanym lub czystym, może zachodzić konieczność odblokowania lub ustawienia i wysłania własnych parametrów - używamy w tym celu odpowiednich kontrolerek. Ważna jest zgodność partycji z adresem startowym zewnętrznego RAM-u, ustawionym w opcjach linkera - w przeciwnym razie program nie zadziała.

O ile wszystko przebiegło pomyślnie (bez zgłaszania błędów transmisji), możemy zamknąć konfigurację i wysłać program, dwukrotnie klikając ikonę lewym przyciskiem myszy. Wyświetlony pasek postępu pokaże przebieg ładowania. Zrezygnowałem z kontroli poprawności ze względu na znaczne spowolnienie całej operacji - po prostu obserwujemy działanie programu po przeładowaniu, a dotychczasowa praktyka wykazała, że takie uproszczone rozwiązanie nie sprawia problemów. Port szeregowy jest zajmowany tylko w chwilach komunikacji z modułem, możemy więc uruchamiać dodatkowe programy powiązane z naszym urządzeniem bez zamykania loader.exe.

Aplikacja została napisana w Delphi 3. Całość - łącznie z pli-



Rys. 6. Rozmieszczenie elementów na płytce drukowanej.

kami źródłowymi oraz procedurą obsługi portu szeregowego - jest zamieszczona na stronie EP w dziale „Download“ jako free-ware.

Programowanie

Duża pamięć programu, a zwłaszcza znaczna pamięć danych pozwalają na swobodne stosowanie w programowaniu DS5000 języków wysokiego poziomu - przede wszystkim C. Przez wiele lat główną przeszkodą były wysokie ceny legalnych kompilatorów.

Obecnie dostępny jest freeware'owy kompilator SDCC (Small Devices C Compiler), rozpowszechniany na zasadach General Public License. Jego autor, Sandeep Dutta, stworzył go głównie z myślą o środowisku Linux. SDCC można jednak z powodzeniem używać również w systemach Windows 9x. Wszelkie niezbędne programy źródłowe i dokumentacje są dostępne na stronie <http://sdcc.sourceforge.net>. Jeśli nie chcemy samodzielnie kompilować plików źródłowych - mamy tam też do pobrania pliki wykonywalne. Są w dodatku dostępne dwie wersje pakietu dla Windows:

1. Utworzona za pomocą bezpłatnego kompilatora C Cygwin. Nie może pracować samodzielnie, gdyż korzysta z bibliotek Cygwin. Wymaga to ściągnięcia i instalacji całego Cygwin albo przynajmniej potrzebnych komponentów.

2. Utworzona za pomocą kompilatora Borland C++. Jest samodzielna oraz (według napotkanych przeze mnie opinii) szybsza w działaniu. Na razie nie posiada jednak debuggera (który występuje w wersji Cygwin).

Każda z wersji jest kompilatorem uruchamianym konsolowo z linii komend. Nie jest to niestety zbyt wygodne - zwłaszcza obecnie przy zaawansowanych i przyjaznych środowiskach graficznych. O ile nawet usprawnimy sobie pracę za pomocą plików wsadowych, to głównym mankamentem pozostaje analiza raportu błędów i mozolne odszukiwanie ich w kodzie źródłowym.

Dlatego rozpocząłem składanie takiego uproszczonego, amatorskiego środowiska dla Windows z użyciem Delphi. Pomija ono z założenia wiele elementów znanych z pakietów profesjonalnych. Jego głównym celem nie jest tworzenie bardzo rozbudowanych aplikacji, ale ułatwienie i uprzyjemnienie amatorskiego pisania średniozaawansowanych programów w C. Ekran środowiska jest przedstawiony na rys. 7.

Jako edytor został wykorzystany znakomity program TSynEdit. Jest on również dostępny na zasadach Mozilla Public License lub General Public License. Umożliwia m.in.: bardzo sprawne i niezawodne kolorowanie składni, wszystkie typowe operacje edycyjne, wielopoziomowe cofanie UNDO/REDO, sprawne wyszukiwanie i zastępowanie, dowolny wybór czcionki, zaznaczenie zakładek na marginesie i wiele innych operacji. Wszelkie informacje, pliki źródłowe, gotowe pakiety Delphi, programy demonstracyjne są dostępne na stronie <http://synedit.sourceforge.net>.

Wykonawczym kompilatorem jest SDCC w wersji „Borland“. Na tym etapie brak debug-

gera nie jest kłopotem - przygotowanie jego obsługi w środowisku graficznym wymaga jeszcze pracy. Na marginesie należy stwierdzić, że o ile debugger znakomicie pomaga w sprawdzaniu przeliczeń, konwersji itp., to przy uruchamianiu rzeczywistego układu z różnymi transmisjami, przezwaniami i zależnościami czasowymi bardzo często staje się mało przydatny. To jest wprawdzie subiektywna opinia, ale przy pewnej wprawie częstokroć dokładna obserwacja zachowania się programu plus kilka sygnałów testowych przynoszą szybsze rozwiązania niż mozolne śledzenie zawartości rejestrów przy pracy krokowej.

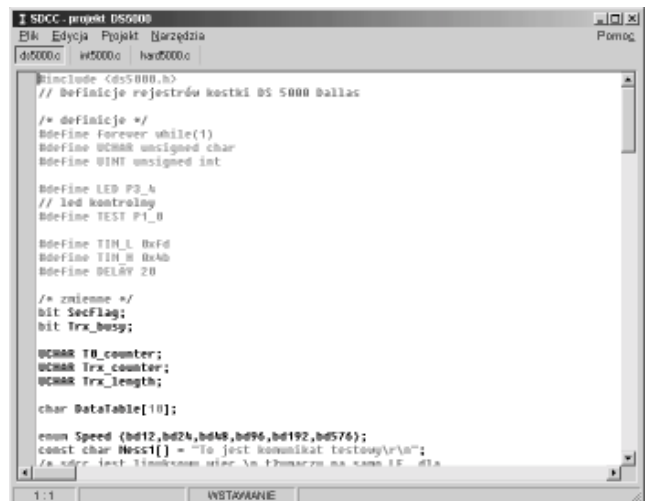
Na razie środowisko posiada następujące możliwości:

1. Operacje plikowe: otwieranie, zamykanie i zapisywanie plików. Dotyczą one zawsze pliku w aktualnie widocznej zakładce edytora.

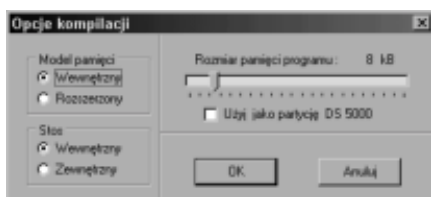
2. Operacje na całych projektach: otwieranie, zamykanie i zapisywanie, dodawanie i usuwanie plików, ustawianie opcji.

Projekt obejmuje wykaz używanych plików oraz konfigurację kompilatora. Zrezygnowałem z typowego oddzielnego okienka opisujuącego otwarty projekt - wszystkie pliki projektu są pokazane jako zakładki edytora. Dodanie lub usunięcie pliku jest równoznaczne z dodaniem lub usunięciem zakładki.

3. Kompilacja plików *.c i *.asm do postaci relokowalnej *.rel albo wynikowej *.ihx (Intel Hex). SDCC dopuszcza jednorazo-



Rys. 7. Ekran windowsowego środowiska dla SDCC.



Rys. 8. Widok okna ustawiania konfiguracji kompilatora.

wo kompilację tylko pojedynczego pliku. Jeśli chcemy złożyć projekt z kilku modułów, musimy wcześniej dodatkowo moduły skompilować do postaci relokowalnej **.rel*, tak żeby linker miał je już gotowe. Dodatkowo na pierwszej pozycji wywołania musi się znaleźć moduł funkcji *main()*. Te ograniczenia są na ekranie odzwierciedlone następująco:

- na pierwszej zakładce musi być ulokowany plik **.c* z funkcją *main ()*;
- komenda utworzenia pliku wynikowego **.ihx* jest aktywna tylko dla pierwszej zakładki, wszystkie następne zakładki projektu umożliwiają tylko utworzenie **.rel*.

Obecnie wszystkie potrzebne pliki **.rel* należy utworzyć (lub aktualizować) samodzielnie. Brak pliku **.rel* lub jego aktualizacji (tj. ponownej kompilacji po zmianach wprowadzonych w kodzie) jest zaznaczany podkreśleniem nazwy pliku na zakładce, w takiej sytuacji tworzenie **.ihx* zostaje wstrzymane z odpowiednim komunikatem. Jest to rozwiązanie tymczasowe - program jest przygotowany do wprowadzenia typowych operacji *make* i *build* (choć przy niezbyt „rozdrobnionych” projektach doskonale można się bez nich obywać).

4. Ustawianie opcji kompilacji dla projektu. W tej chwili są to tylko najbardziej ogólne ustawienia (okienko opcji na **rys. 8**):

- model pamięci (wewnętrzny/small - domyślną lokacją zmiennych jest obszar DATA, rozszerzony/large - zmienne są domyślnie lokowane w XDATA; autor SDCC zaleca stosowanie

small+jawne deklaracje lokacji zmiennych w razie potrzeby),

- stos (wewnętrzny - w obszarze IDATA, zewnętrzny - pierwsze 256 bajtów w obszarze XDATA, stosowanie zewnętrznego stosu daje większą swobodę, ale kosztem szybkości),
- rozmiar pamięci programu (używany do alarmowania o przekroczeniu pojemności zastosowanego mikrokontrolera,
- ustawienie adresu startowego RAM za obszarem programu - opcja specjalnie dla DS5000.

5. Lokalizacja błędów. W razie wystąpienia błędów kompilacji ich wykaz jest pokazywany w oddzielnym oknie. Podwójne kliknięcie na linii opisu błędu powoduje przejście do odpowiedniego miejsca w kodzie programu (**rys. 9**). Mechanizm raportowania błędów SDCC nie zawsze działa prawidłowo - jeśli nie jest podany numer wiersza, samoczynna lokalizacja nie jest możliwa i trzeba ręcznie odszukać błąd, korzystając z komendy *Szukaj* edytora.

W najbliższych planach rozwojowych programu przewiduje się wyposażenie go w:

- menedżera bibliotek,
- menedżera i edytor plików nagłówkowych,
- ładowanie programu do symulatorów (EPROM, DS5000) z poziomu środowiska.

Cały projekt w Delphi (wraz z plikami źródłowymi) jest do-

WYKAZ ELEMENTÓW

Rezystory

R1: 4,7k Ω

Kondensatory

C1: 100nF

C2, C3: 33pF

C4: C7: elektrolit 10 μ F/25V

Półprzewodniki

U1: DS5000(T)

U2: MAX232 (lub odpowiednik)

U3: 4052

U4: 4066

Różne

X1: 11,059MHz

JP 1: jumper

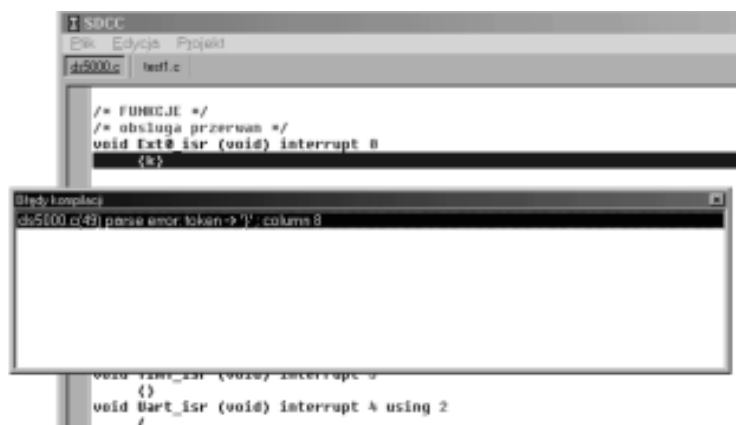
J1: gniazdo DB9M do druku

Z1: gniazdo pionowe 40 pin
płytką drukowaną dwuwarstwowa taśma 40-żyłowa z wtykiem 40 pin i wtykiem testowym DIL40

stępnym jako freeware na zasadach Mozilla Public License/General Public License. Można go pobrać ze strony Elektroniki Praktycznej. Tam też znajduje się dodatkowy plik opisu (gdyż żadna pomoc na razie jeszcze nie powstała) oraz dokumentacja wspomnianych licencji.

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

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/pazdziernik01.htm> oraz na płycie CD-EP10/2001B w katalogu PCB.



Rys. 9. Sposób lokalizowania błędów w SDCC.