

Emulator DS5000 i innych mikrokontrolerów rodziny '51, część 1

AVT-5029



Firma Dallas Semiconductor (USA) jest znana z opracowania wielu mikrokontrolerów opartych na architekturze 8051, ale znacznie udoskonalonych i wyposażonych w różne dodatkowe funkcje użytkowe. W artykule zajmiemy się układami nieco starszymi. Są one określane przez producenta wspólną nazwą „Soft microcontrollers“ i obejmują kilka typów układów, z których najprostszym jest DS5000.

Układ DS 5000 był już dosyć dawno opisywany skrótowo na łamach Elektronika Elektora oraz Elektroniki Praktycznej, ale bez przykładów praktycznego zastosowania. Obecnie proponujemy ze-

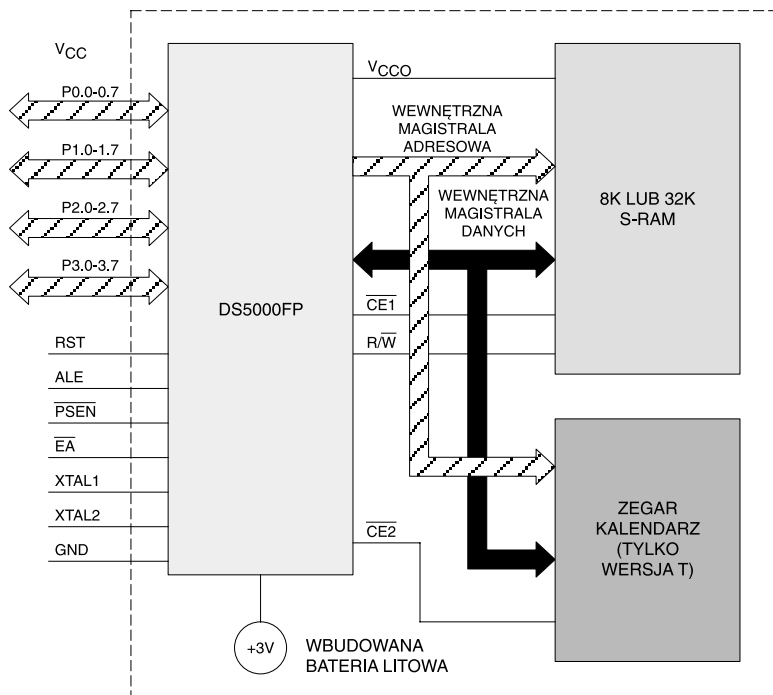
staw sprzętowo-programowy oparty właśnie na układzie DS5000. Rozpocniemy od przypomnienia niektórych (bo pełne opisy zajmują kilkadziesiąt stron podręcznika), najważniejszych właściwości tego mikrokontrolera.

Skrócona charakterystyka DS5000

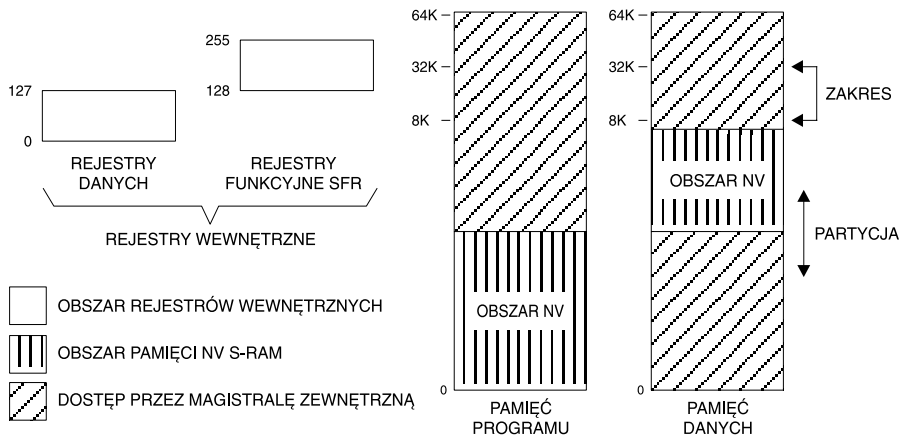
DS5000 jest wbudowanym w obudowę DIL40 (tylko wyższą niż typową) kompletnym, 8-bitowym systemem mikroprocesorowym zawierającym (schemat blokowy układu przedstawiono na rys. 1):

- jednostkę centralną DS5000FP,
- pamięć SRAM 8kB lub 32kB (w zależności od wersji),
- litową baterię podtrzymującą o trwałości 10 lat,
- dodatkową wewnętrzną magistralę dostępu do pamięci,
- w wersji „T“ układu - zegar/kalendarz DS1215 z dostępem szeregowym.

Mikrokontroler DS5000FP jest całkowicie zgodny z mikrokontrolerami serii '51. Nie wymaga dodatkowych rozkazów, posiada wszystkie funkcje i rejestry '51, takie same są również parametry elektryczne linii wejść i wyjść. Ponieważ w tym mikrokontrolerze zapewniono identyczny rozkład wyprowadzeń jak w układach serii



Rys. 1. Ogólny schemat blokowy układu DS5000 (T).



Rys. 2. Mapa pamięci DS5000.

'51, możemy we wszystkich projektach płytek stosować dotychczasowe rozwiązania używane przy '51 czy wręcz zamienić w układzie dotychczasowy mikrokontroler na nowy.

Jedynie różnice dotyczą pinu EA, na który należy podać poziom wysoki, aby korzystać z pamięci wewnętrznej oraz pinu RST, który można pozostawić nie podłączony. Zauważmy, że zastosowanie dodatkowych, wewnętrznych magistral dostępu do pamięci nie zajmuje portów P0 i P2 przy realizacji rozkazów MOVX i MOVC - porty te można swobodnie wykorzystywać do innych celów. Pozostawiono także możliwość pracy tradycyjnej (EA = LOW) z pamięciami zewnętrznymi i multipleksowaną magistralą opartą na P0 i P2, ale praktycznie jest to mało użyteczne, bo tracimy większość korzyści związanych ze scaleniem układu.

Wbudowany SRAM jest podzielony na dwie partycje: programu i danych (ich rozmiary można swobodnie zmieniać). Podczas normalnej pracy partycja programu umożliwia tylko odczyt, pełny dostęp mamy jedynie do partycji danych (tak jak w rozwiązaniach tradycyjnych). Kod programu jest wpisywany podczas procesu programowania. Zwróćmy uwagę, że podtrzymanie bateryjne obejmuje oczywiście obie partycje, zatem cały obszar danych jest nieulotny i nie ma potrzeby stosowania pamięci EEPROM dla stałego lub okresowego przechowania danych.

Jak widać na rys. 2, przestrzeń adresowa obejmuje całe 64kB obsługiwane przez '51. Jednak w we-

wnętrznej pamięci nieulotnej mamy do dyspozycji obszary ograniczone:

- dla kodu programu - od adresu 0 do granicy podziału partycji,
- dla danych - od granicy podziału partycji do ustawionego zakresu (dla kostki 32kB można wybrać zakres 8kB albo 32kB).

Adresowanie kodu programu lub danych poza powyższymi granicami powoduje samoczynne odwołanie do pamięci zewnętrznej przez porty P0 i P2. DS5000FP posiada w porównaniu z podstawowym mikrokontrolerem '51 wiele dodatkowych udogodnień i zabezpieczeń. Należą do nich:

1. Wbudowany układ zerowania przy włączaniu zasilania - nie zachodzi konieczność używania elementów zewnętrznych.
2. Dodatkowa flaga w rejestrze PCON określająca przyczynę zerowania (włączenie zasilania, zerowanie programowe albo wygenerowane przez watchdoga) - znacznie ułatwia napisanie bardziej rozbudowanych procedur inicjalizacji procesora.
3. Wbudowany układ watchdoga.
4. Szereg zabezpieczeń przed spadkiem lub zanikiem napięcia zasilającego:

- dodatkowe przerwanie wywołane spadkiem napięcia poniżej określonej wartości,
- blokada zapisu do pamięci przy spadku napięcia zasilającego,
- rozbudowany mechanizm przełączania podtrzymania baterijnego gwarantujący eliminację błędnych operacji i stabilność danych w pamięci i rejestrach.

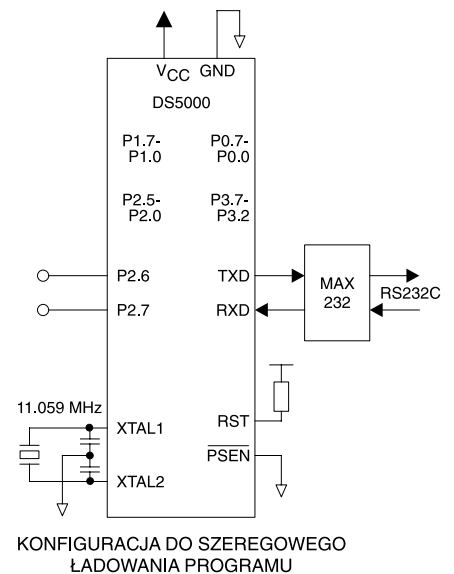
5. Zabezpieczenia przed nieautoryzowanym odczytem kodu:

- bit blokady odczytu,
- szyfrowanie danych i adresów oparte na 40-bitowym kluczu kodowym,
- dodatkowe puste cykle dostępu do pamięci wstawiane losowo pomiędzy cykle użytkowe,
- dodatkowy ukryty obszar 48 bajtów pamięci RAM (Vector RAM) służący do przechowania wektorów przerwań w trybie pracy szyfrowanej (uniemożliwia to zlokalizowanie zaszyfrowanego adresu obsługi przerwan).

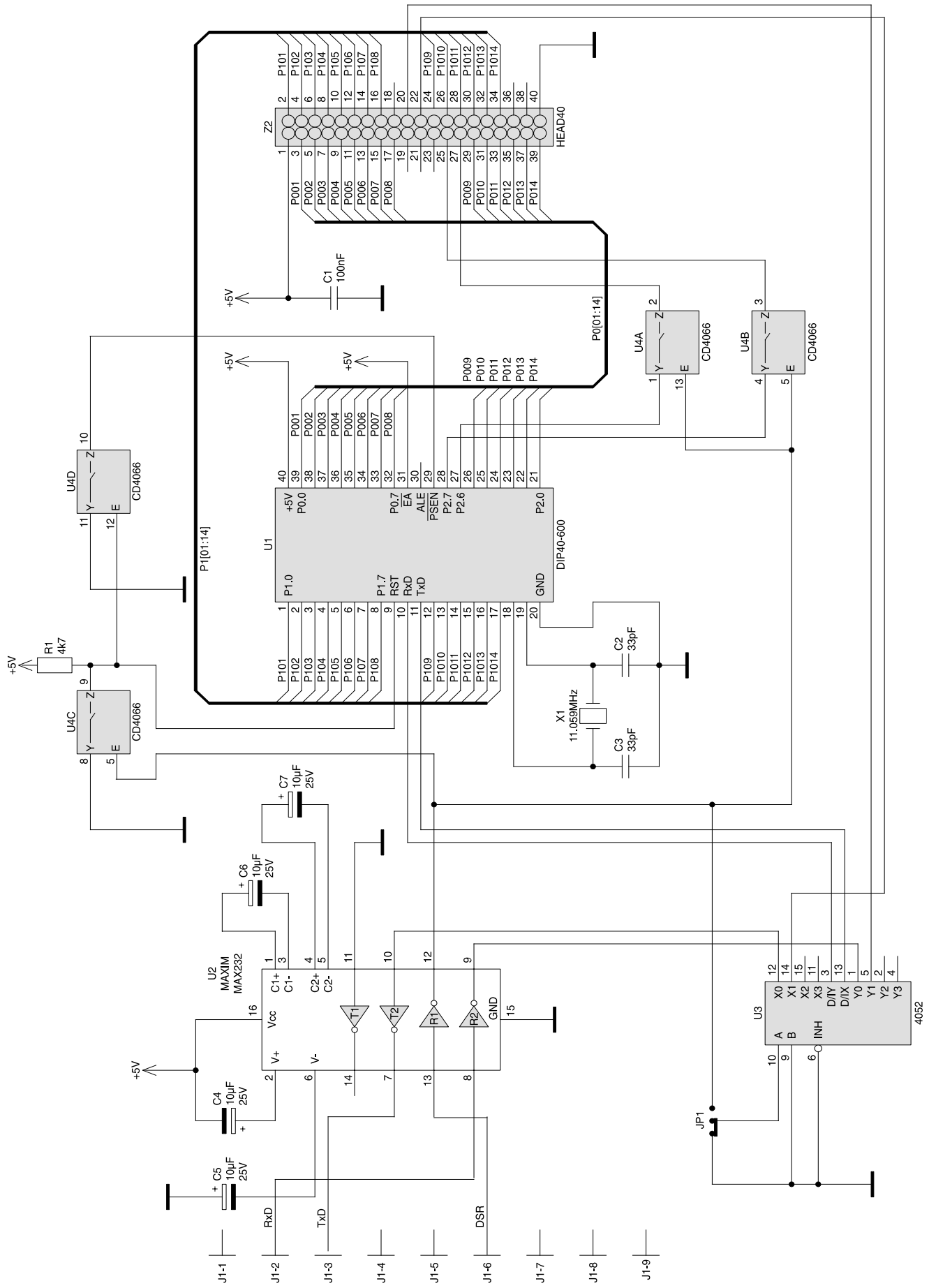
6. Mechanizm ograniczonego czasowo dostępu do najważniejszych flag i rejestrów (np. decydujących o partycjonowaniu pamięci czy obsłudze watchdoga). Znacznie ogranicza to możliwość przypadkowego przełączenia opcji istotnych dla pewności działania programu.

7. Wbudowany mechanizm ładowania kodu do NV-RAM (bootloader) wykorzystujący port szeregowy i całkowicie niewidoczny dla programu użytkowego (nie zajmuje żadnego obszaru pamięci kodu ani danych). Aktywacja bootloadera następuje po odpowiednim ustawieniu stanu wejść (rys. 3) i wysłaniu portem szeregowym znaku #13 (CR). Podczas odbioru CR jest też samoczynnie rozpoznawana szybkość transmisji (zakres dozwolonych szybkości zależy od zastosowanego kwarcu).

Jak widać, podłączenia są bardzo proste - łatwo je zrealizować



KONFIGURACJA DO SZEREGOWEGO ŁADOWANIA PROGRAMU
Rys. 3. Konfiguracja uruchomienia boot-loadera.



Rys. 4. Schemat elektryczny emulatora.

w układzie docelowym, co zresztą producent zaleca. Należy jedynie pamiętać o utrzymaniu poziomu wysokiego na P2.6 i P2.7 (poziom niski na tych pinach może być odebrany jako żądanie trybu programowania równoległego, który w układzie jest także dostępny).

Prawidłowo uruchomiony boot-loader zgłasza się komunikatem tekstowym i oczekuje na dalsze komendy (wysyłane także tekstowo). Obejmują one pełny zakres operacji obsługiwanych przez moduł (konfiguracja pamięci, ustawianie zabezpieczeń, ładowanie pliku Intel Hex, odczyty kontrolne itd.). Wszelkie szczegóły dotyczące używania modułu - tutaj z konieczności pominięte - są dostępne w dokumentacjach na stronie producenta (www.dalsemi.com). Powyższe skrótkowe omówienie pokazuje, że nawet dzisiaj - przy ekspansji nowych układów z pamięciami flash - DS5000 pozostaje silnym mikrokontrolerem ułatwiającym projektowanie urządzeń (zwłaszcza przeznaczonych do niezawodnej autonomicznej pracy).

Przykład zastosowania

Podobnie jak dla większości bardziej złożonych układów, dla DS5000 istnieje jego dodatkowy osprzęt wspomagający także poznawanie i programowanie. W witrynie www firmy Dallas można znaleźć bezpłatny program obsługi wspomnianego powyżej boot-loadera, można też zakupić kit uruchomieniowy (DS5000K). Jednak jego cena jest dosyć wysoka.

Dlatego proponujemy wykonanie takiego zestawu uruchomieniowego w uproszczonej wersji, o nieco ograniczonych możliwościach. Może on służyć jako symulator zarówno układu DS5000(T), jak i całej serii '51 z wewnętrzną pamięcią programu (8751, 89c51, częściowo 89c52, zaś po zastosowaniu dodatkowego adaptera 89c1051, 2051 i 4051). Wspomnianym ograniczeniem jest możliwość zastosowania pojedynczego kwarcu o częstotliwości tylko 11,059MHz (optymalnej z punktu widzenia transmisji, ale nie zawsze zgodnej z potrzebami uruchamianego urządzenia).

Schemat układu przedstawiono na **rys. 4**. W zaprojektowanym układzie realizowane jest aktywacja boot-loadera za pomocą możliwie prostych środków. Układ U1 to DS5000 lub 5000T w podstawce DIL40. Układ U2 (MAX 232) zapewnia konwersję poziomów napięć TTL<>RS232C.

Klucze analogowe 4066 (U4) ustawiają wymagane poziomy (w zależności od stanu linii DTR nadrzędnego komputera sterującego) i zapewniają odłączenie linii P2.6 oraz P2.7. Multiplexer 4052 (U3) służy do przełączania linii komunikacyjnych RxD i TxD pomiędzy komputerem i obsługiwanym urządzeniem. Każdorazowe włączenie boot-loadera powoduje dołączenie portu szeregowego do U2. Przy pracy programu port może być podłączony do uruchamianego urządzenia albo do U2 (w zależności od położenia JP1). Druga możliwość jest użyteczna, jeśli nasz program ma się również komunikować z własnym programem narzędziowym na PC.

Od strony komputera podłączamy się przez typowe gniazdo DB9M. Od strony urządzenia zastosowano pionowe gniazdo 40 pin z łapkami. Można tam podłączyć taśmę z wtykiem testowym DIL40 albo adapter do 89cxx51. Całość jest zasilana z uruchamianego urządzenia - należy zawsze dokładnie sprawdzić położenie wtyku testowego, aby przy odwrotnej polaryzacji napięcia zasilającego nie uszkodzić modułu.

Oprogramowanie sterujące

Jak wspomniano wcześniej, Dallas Semiconductor udostępnia bezpłatnie program *kit.exe* do obsługi modułu. Obecnie jest to nowsza wersja, działająca także pod Windows 9x i wyposażona w plik pomocy. Nadal jednak jest to aplikacja konsolowa (czyli uruchamiana z linii komend w okienku DOS). Jako alternatywny, napisano sterownik w wersji typowo windowsowej. Jego zadaniem jest uproszczenie pracy podczas uruchamiania urządzenia. Główną, często powtarzaną czynnością jest wtedy wysyłanie pliku *hex* do modułu. Aby uniezależnić się od środowiska, w którym piszemy



Rys. 5. Widok okna konfiguracji.

i kompilujemy program '51 - polecenie wysłania hex zostało przypisane do ikony (podwójne kliknięcie) w obszarze paska zadań Windows.

Program *loader.exe* nie wymaga specjalnej instalacji - wystarczy go skopiować wraz z towarzyszącym *ds5000.hlp* do wybranego folderu. Ewentualne skróty w menu start lub na pulpicie tworzymy samodzielnie. Po jego uruchomieniu w pasku zadań pojawia się ikona. Prawy przycisk myszy rozwija niewielkie menu (*Konfiguracja, Pomoc, Koniec*). Pozycja „Konfiguracja” otwiera dodatkowe okno, które pokazano na **rys. 5**.

Wybieramy w nim port, szybkość transmisji oraz nazwę pliku, który będziemy przysyłać (szczegóły w pomocy). Wpisy konfigurujące należy wykonać każdorazowo. Zrezygnowałem z zapisywania tych ustawień, gdyż ich wpisywanie nie stanowi wielkiej niedogodności. Zazwyczaj bowiem uruchamiamy program jednorazowo na dłuższy okres poświęcany pracy z mikrokontrolerem - dwa dodatkowe kliknięcia myszą nie stanowią żadnego problemu.

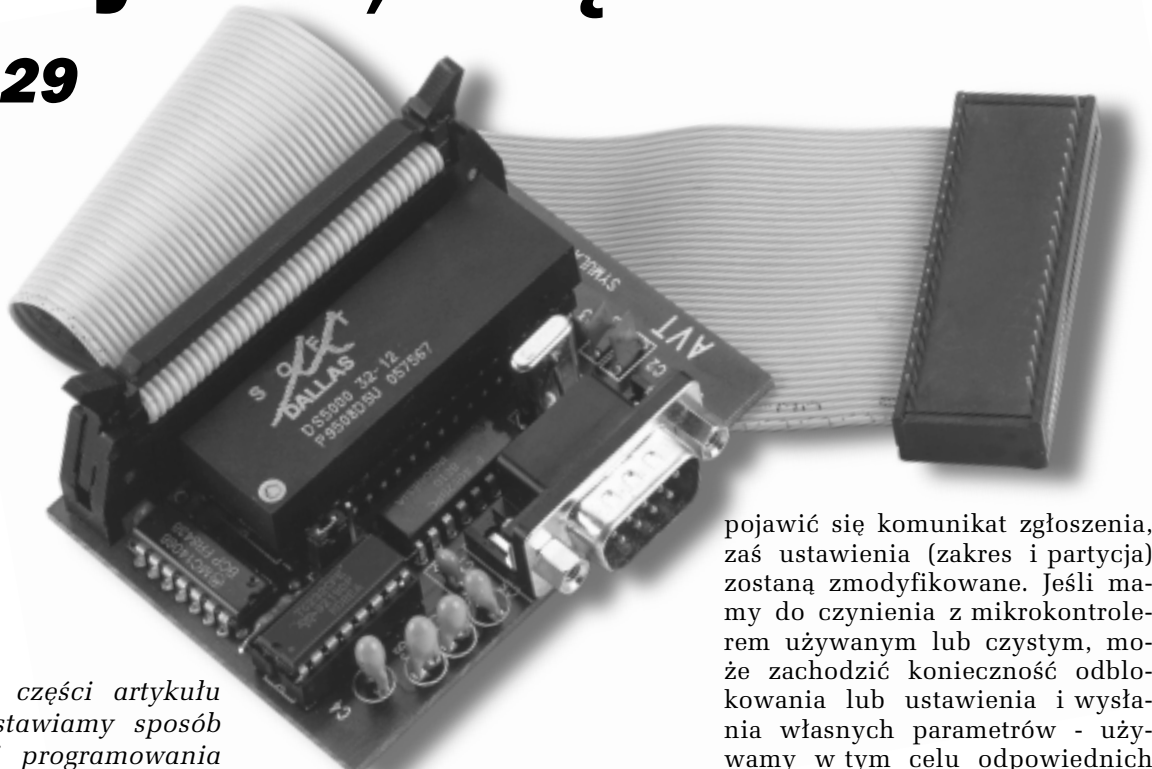
Jak widać, zestaw komend jest maksymalnie uproszczony. Jeśli po zakończeniu pisania chcemy wykorzystać dodatkowe możliwości (np. szyfrowanie), należy skorzystać z oryginalnego *kit.exe*.

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/wrzesien01.htm> oraz na płycie CD-EP09/2001B w katalogu PCB.

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

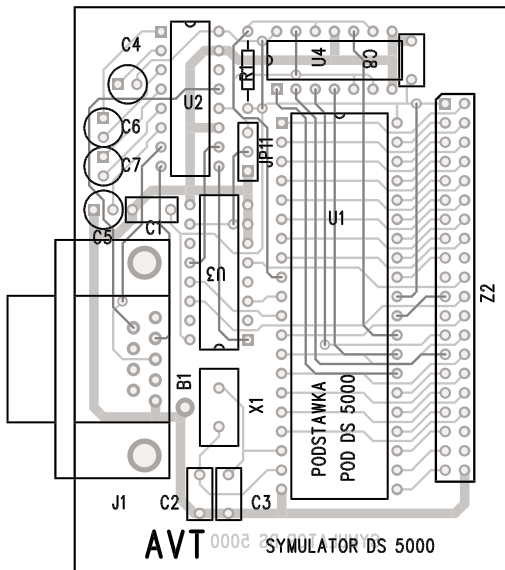
Po złożeniu i wstępnym sprawdzeniu płytki (schemat montażowy na rys. 6) podłączamy ją do wolnego portu szeregowego komputera kablem null-modem (potrzebne są tylko linie GND, TxD, RxD, DTR). Wtyk testowy wkładamy do dowolnej płytki testowej (uwaga na kierunek!). Do prób może to być zwykła podstawka dyl 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 płycie 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.

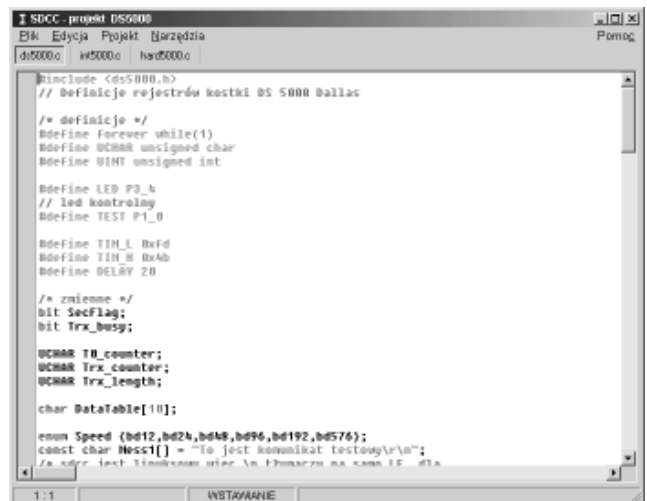
Na razie środowisko posiada następujące możliwości:

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

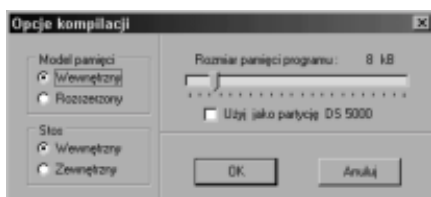
2. Operacje na całych projektach: otwieranie, zamykanie i zapisywanie, dodawanie i usuwanie plików, ustawianie opcji.

Projekt obejmuje wykaz używanych plików oraz konfigurację kompilatora. Zrezygnowałem z typowego oddzielnego okienka opisujującego 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: 4,7kΩ

Kondensatory

C1: 100nF

C2, C3: 33pF

C4: C7: elektrolit 10μF/25V

Półprzewodniki

U1: DS5000(T)

U2: MAX232 (lub odpowiednik)

U3: 4052

U4: 4066

Różne

X1: 11,059MHz

JP 1: jumper

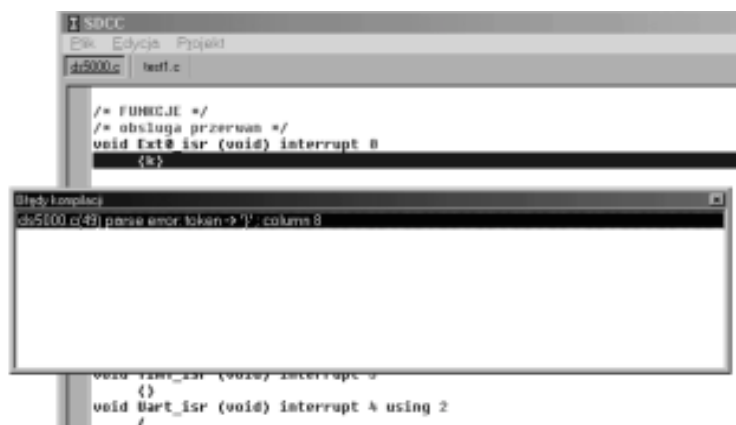
J1: gniazdo DB9M do druku

Z1: gniazdo pionowe 40 pin
płytką drukowaną dwuwarstwową taśmą 40-żyłową 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.