

Programator procesorów ATMEL

Jednymi z najbardziej popularnych obecnie procesorów wykorzystywanych przez amatorów i hobbystów są procesory rodziny 51 produkowane przez firmę Atmel. Procesory te zawierają wewnętrzną pamięć programu Flash, co pozwala wyeliminować zewnętrzną pamięć EPROM, a tym samym znacznie uprościć konstrukcję budowanego urządzenia. Procesory są ogólnie dostępne w sklepach na terenie całego kraju. Aby jednak zmusić je do wykonywania zamierzonych zadań należy je wcześniej zaprogramować. Do tego celu potrzebny jest odpowiedni programator. Programatory dostępne w handlu są zazwyczaj uniwersalne, mają możliwość zaprogramowania wielu typów procesorów, przez co ich koszt jest dość wysoki. Proponuję wykonanie we własnym zakresie prostego i taniego programatora mającego możliwość zaprogramowania/odczytu kilku podstawowych typów procesorów, które są najczęściej wykorzystywane w konstrukcjach amatorskich.

Opisany poniżej programator umożliwia programowania następujących typów procesorów :

AT89C1051	– zawiera 1kB Flash, obudowa DIL 20;
AT89C2051	– zawiera 2kB Flash, obudowa DIL 20;
AT89C4051	– zawiera 4kB Flash, obudowa DIL 20;
AT89C51	– zawiera 4kB Flash, obudowa DIL 40;
AT89C52	– zawiera 8kB Flash, obudowa DIL 40.

Wszystkie powyższe procesory wymagają do zaprogramowania napięcia $V_{pp} = 12\text{ V}$. Dwa ostatnie procesory dostępne są też w wersji z $V_{pp} = 5\text{ V}$. Mają one oznaczenia:

AT89C51-5;
AT89C52-5.

Firma Atmel zapewnia dla swoich procesorów minimum 1000 cykli programowanie/kasowanie. Oznacza to, że każdy procesor powinien dać się zaprogramować co najmniej 1000 razy. Jest to ilość w zupełności wystarczająca do uruchomienia konstruowanego urządzenia.

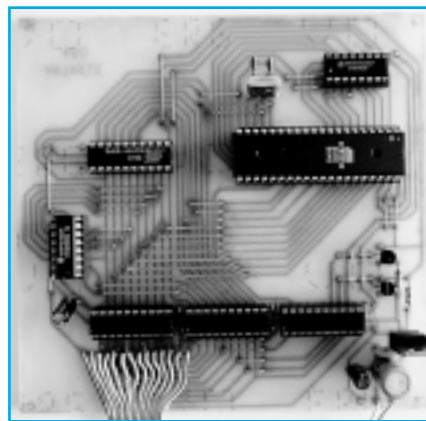
Programator w połączeniu z dobrym programem assemblera jest także doskonałym narzędziem do nauki programowania i konstruowania systemów mikroprocesorowych, a jego koszt jest znacznie niższy od różnego rodzaju dydaktycznych makiet uruchomieniowych.

■ Budowa i zasada działania programatora

Schemat programatora złożony jest z dwóch części. Pierwsza (rys. 1) przedstawia połączenia między elementami elektronicznymi tworzącymi układ programatora, druga zaś (rys. 2) pokazuje sposób połączenia podstawek pod programowane procesory z układem z rysunku 1. Najkorzystniej jest zastosować podstawki precyzyjne, które są tanie oraz trwalsze i wygodniejsze w użyciu niż zwykłe. Można też pokusić się o zastosowanie podstawek z dźwignią.

Programator połączony jest z komputerem IBM-PC przez złącze drukarki przewodem 14-żyłowym o długości ok. 1,5 m (nie jest wymagany port dwukierunkowy). Najlepiej nadaje się do tego celu wstęga przewodów komputerowych lub 14-żyłowy kabel ekranowany zakończony wtykiem typu DB25. Port drukarki posiada 12 linii wyjściowych, przez które przekazywane są sygnały sterujące do układu programatora. Cały programator złożony jest z 5 cyfrowych układów scalonych i paru elementów dyskretnych.

Układy US1, US2, US3 stanowią 8-bitowe rejestry-zatrzaski, których wejścia są połączone ze sobą i dołączone do 8 linii wyjściowych złącza drukarki (gniazdo G1 nóżki 2÷9). Wpis danych do rejestrów odbywa się przez podanie stanu wysokiego na wejście C – wtedy stany logiczne z wejść



D1÷D8 rejestrów przepisywane są na ich wyjścia Q1÷Q8. Gdy na wejściu C rejestrów jest poziom niski, to zmiana stanów na wejściach D1÷D8 nie powoduje zmiany stanów na wyjściach Q1÷Q8 - rejestry pamiętają dane.

Wejścia C sterujące wpisem i pamiętaniem danych sterowane są przez trzy linie złącza drukarki. Ostatnia wolna linia tego złącza jest dołączona do wejścia OC układu US1. Jeśli wystąpi na niej wysoki poziom logiczny, to wyjścia Q1÷Q8 układu US1 przejdą w stan wysokiej impedancji. W przeciwnym przypadku na tych wyjściach będą panowały określone stany logiczne wpisane wcześniej wysokim stanem na wejściu C układu US1.

Wyjścia Q1÷Q8 rejestru US1 połączone są z wejściami procesora, przez które podawane są kolejne bajty do zaprogramowania, oraz z wejściami układu US4, który jest 8-bitowym rejestrem przesuwającym z wejściami równoległymi i szeregowym wyjściem. Wyjście rejestru przesuwającego połączone jest z linią wejściową złącza drukarki. Umożliwia to odczyt-weryfikację każdego zaprogramowanego bajtu danych i wykrycie ewentualnych błędów programowania, bądź też odczyt kodu programu znajdującego się w procesorze i jego zapis do pliku w komputerze.

Zastosowanie szeregowego rejestru przesuwającego jest konieczne za względu na to, że złącze drukarki (szczególnie jednokierunkowe) posiada tylko 5 linii wyjściowych, a dla odczytu jednego bajtu (8 bitów) potrzebne jest 8 wejść. Takie rozwiązanie umożliwia też zaoszczędzenie kilku dodatkowych żył przewodu połączeniowego pomiędzy układem programatora, a komputerem.

Podczas odczytu kodu programu zapisanego w procesorze wyjścia układu

US1 znajdują się w stanie wysokiej impedancji, aby zapobiec zwarciom między sygnałami z procesora, a sygnałami z wyjść Q1 ÷ Q8 układu US1.

Wpis 8 bitów danych do rejestru przesuwającego US4 następuje podczas podania niskiego stanu logicznego na wejście SH/LD rejestru. Przesuw danych w rejestrze o 1 bit odbywa się po podaniu narastającego zbocza sygnału na wejście CLK US4. Podając kolejno 8 narastających zboczy na wejście CLK US4 uzyskujemy pojawianie się na wyjściu QH US4 kolejnych bitów danych w kolejności od najstarszego do najmłodszego. Bity te są odczytywane przez komputer i składane w 1 bajt.

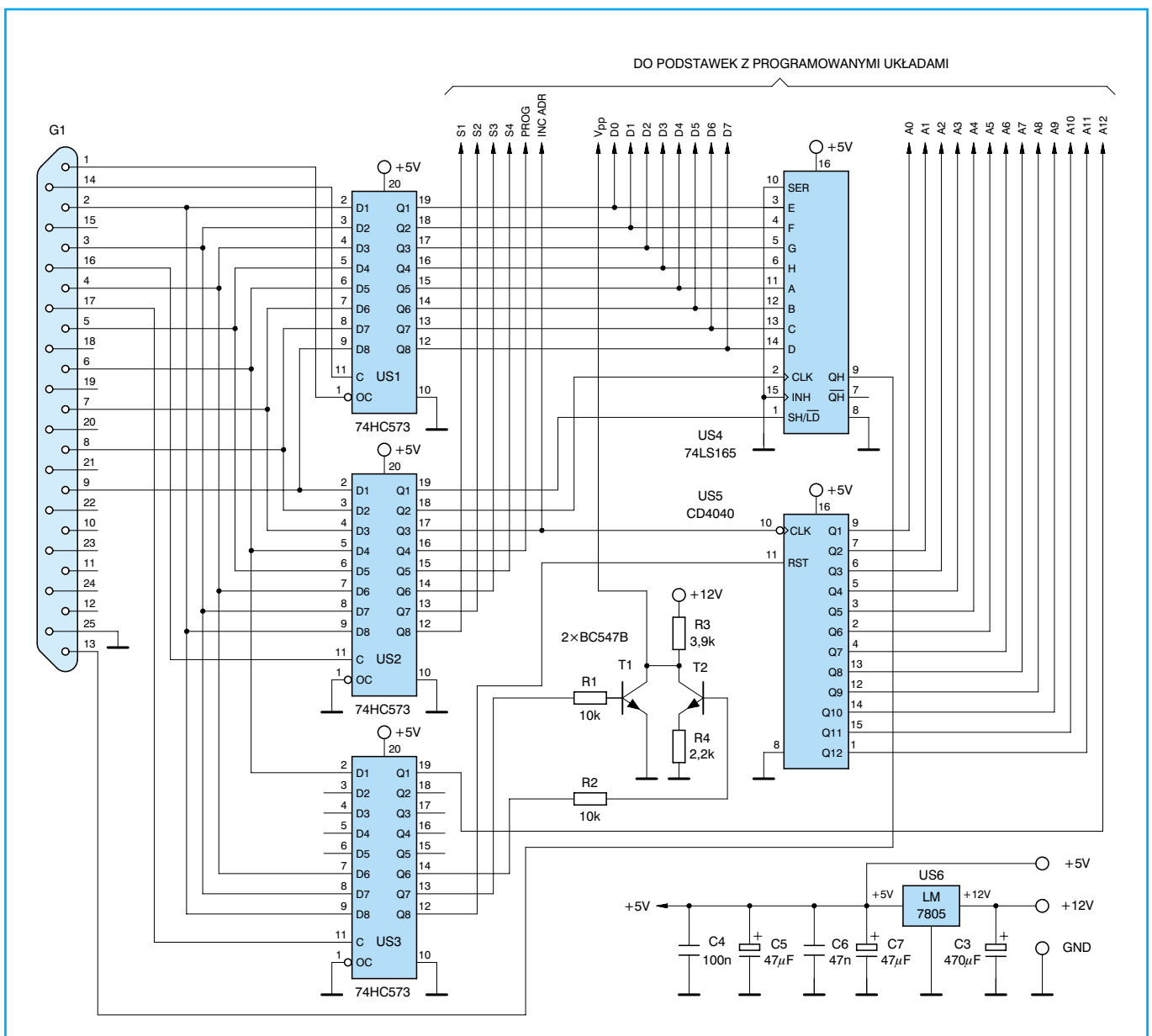
Wejścia sterujące CLK i SH/LD rejestru przesuwającego są połączone z wyj-

ściami układu US2, który podaje także sygnały sterujące na wejścia procesora. Odpowiednia kombinacja stanów logicznych na liniach S1, S2, S3, S4 określa tryb pracy procesora – odczyt kodu programu, odczyt sygnatury procesora, programowanie danych, programowanie bitów zabezpieczających przed odczytem. Linia INC ADR służy do zwiększania licznika adresów wewnętrznej pamięci Flash.

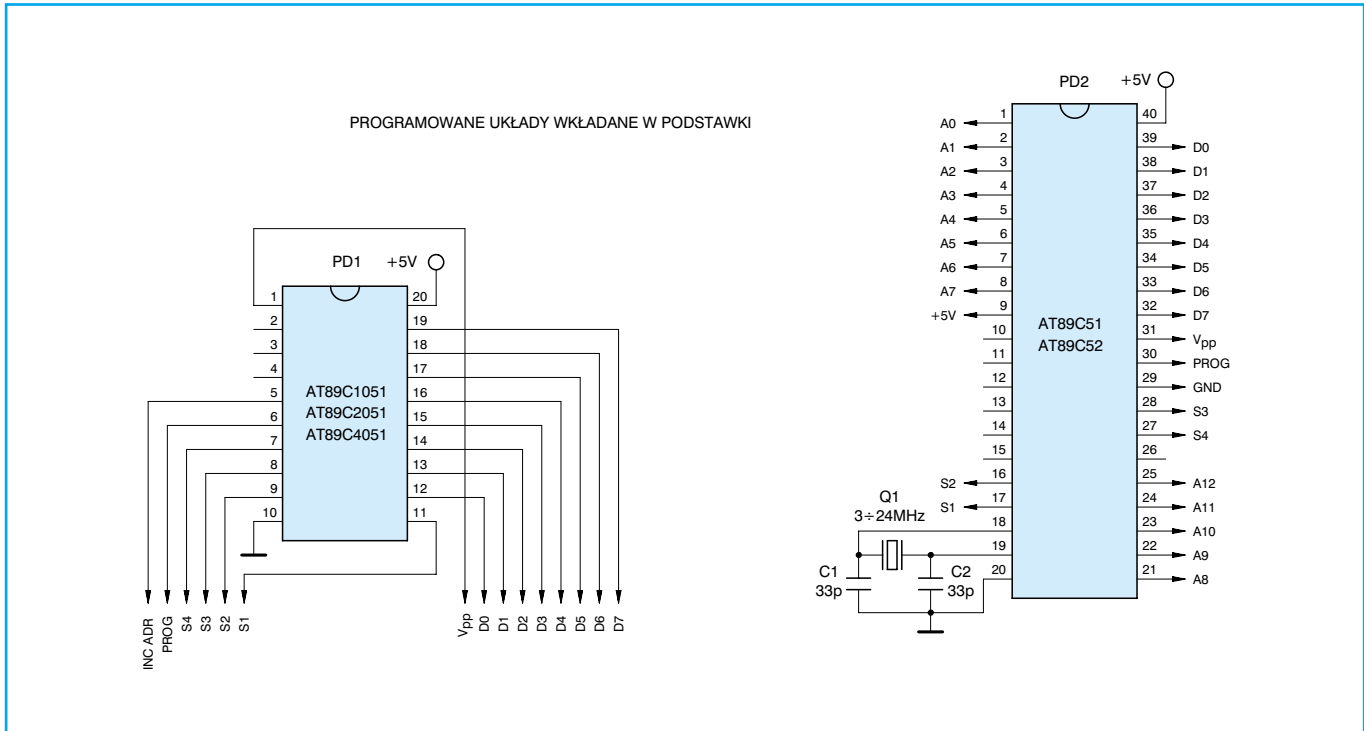
Procesory w obudowach 20-końcówkowych, DIP 20 (AT89C1051, 2051, 4051) mają wbudowany licznik adresów, zaś procesory w obudowach 40-końcówkowych, DIP 40 (AT89C51, 52) wymagają podania adresu przez wyprowadzenia zewnętrzne. Z tego też względu zastosowano układ US5, bę-

dący 12-bitowym licznikiem binarnym. Jego wejście zliczające CLK dołączone jest do linii INC ADR. Wyjścia licznika US5 dołączone są do wejść adresowych programowanego lub odczytywanego procesora w obudowie 40-końcówkowej.

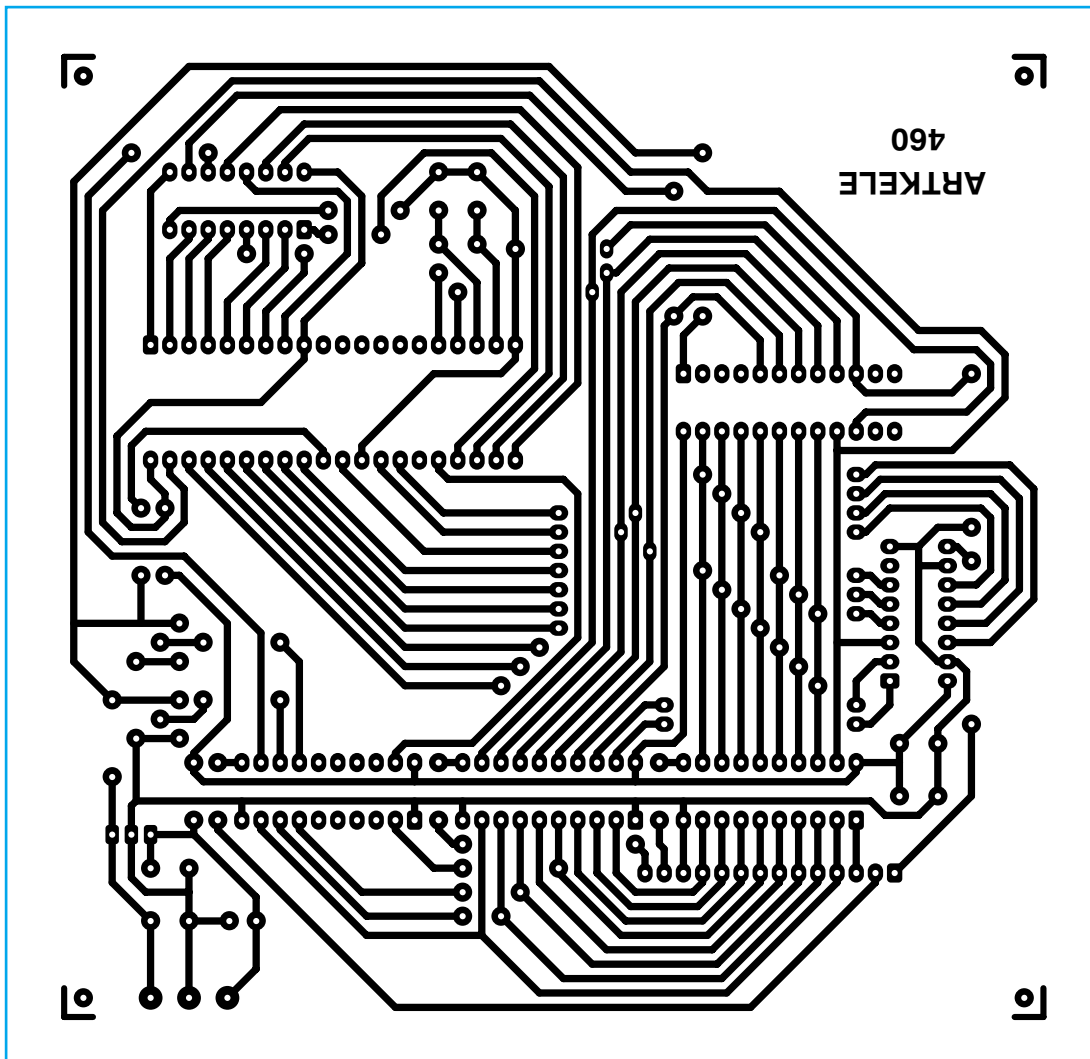
Procesor AT89C51 posiada 4 kB wewnętrznej pamięci programu Flash, co wymaga użycia 12 linii adresowych ($2^{12} = 4096$), a więc dokładnie tyle ile posiada licznik. Pojemność pamięci w procesorze AT89C52 jest jednak dwukrotnie większa - wynosi 8 kB. Wymaga to zastosowania jednej dodatkowej linii adresowej. Linia ta wyprowadzona jest z jednego z wyjść rejestru US3, z którego podawany jest też sygnał zerowania licznika US5.



Rys. 1 Schemat ideowy programatora



Rys. 2 Schemat podłączenia podstawek pod programowane procesory



Rys. 3a Płytką drukowaną i rozmieszczenie elementów

Linia PROG służy do podania krótkiego impulsu programującego, który następuje po uprzednim ustawieniu sygnałów S1, S2, S3, S4 na tryb programowania, ustawieniu odpowiedniego adresu, bajtu danych oraz napięcia programującego V_{pp} . Linia V_{pp} może przyjmować trzy wartości napięcia: 0,5 lub 12 V, aby zapewnić możliwość zerowania, odczytu i programowania procesora. Ustawienie odpowiedniej wartości napięcia następuje za pomocą dwóch tranzystorów T1 i T2 sterowanych z rejestru US3 (przez rezystory ograniczające prąd bazy R1 i R2).

Podanie zera logicznego na bazę tranzystora powoduje jego zatkanie, zaś podanie jedynki logicznej wprowadza tran-

Program obsługujący układ programatora o nazwie „Progat” napisany jest w Turbo Pascalu 7.0. Posiada on następujące opcje:

- odczyt kodu programu zapisanego w procesorze i jego zapis do pliku o podanej nazwie w formacie Intel-hex;
- zaprogramowanie procesora danymi z pliku o podanej nazwie (dane w pliku muszą być zapisane w formacie Intel-hex);
- zaprogramowanie bitów zabezpieczających w procesorze, czyli uniemożliwienie późniejszego odczytu programowanych danych (każdy bajt odczytany z zabezpieczonego procesora będzie miał wartość 255, czyli same jedynki);
- wybór adresu złącza drukarki, do którego podłączony jest programator, możliwe ustawienie adresów: 278H, 378H oraz 3BCH (na karcie graficznej Hercules);
- ustawienie opóźnienia powodującego spowolnienie działania programu, co jest przydatne dla komputerów z zegarem o częstotliwości powyżej 300 MHz (przy braku opóźnienia układ elektroniczny programatora może nie nadążać za zmianami sygnałów na porcie drukarki).

Wybór poszczególnych opcji odbywa się przez uruchomienie programu z odpowiednim parametrem. Uruchomienie programu bez podania nazwy pliku lub opcjonalnie parametru spowoduje wyświetlenie informacji dotyczących sposobu właściwego wywołania programu. Sam plik programu „progat.exe” nie powinien mieć ustawionego atrybutu „tylko do odczytu”, gdyż podczas zmiany adresu portu drukarki lub opóźnienia, nowe wartości zapamiętywane są w tym właśnie pliku.

Przy programowaniu lub odczytywaniu danych za każdym razem tworzony jest plik o nazwie „progat.bin”. Zawiera on kod binarny danych odpowiadających danym w formacie Intel-hex w odczytywanym lub utworzonym pliku. Podczas operacji programowania dokonywana jest weryfikacja po każdym zaprogramowanym bajcie. Polega ona na odczytaniu ostatniego zaprogramowanego bajtu i porównaniu go z bajtem, który był programowany. Jeśli te wartości są różne, to następuje przerwanie programowania i wyświetlenie komunikatu o błędzie. W rzeczywistości

program po stwierdzeniu błędu próbuje jeszcze dwukrotnie zaprogramować i zweryfikować dany bajt, zanim definitywnie przerwie operację programowania. Pozwala to wyeliminować przypadkowe błędy jakie mogą wystąpić w transmisji danych, spowodowane np. zakłóceniami zewnętrznymi.

Czas programowania uzależniony jest głównie od ilości programowanych bajtów kodu programu. Zaprogramowanie całego obszaru pamięci dostępnej w procesorze trwa kilka sekund. Program obsługujący wyświetla na bieżąco ilość zaprogramowanych danych w bajtach oraz w procentach. Bajty o wartości OFFH nie są programowane, gdyż podczas operacji kasowania wszystkie bajty w pamięci Flasch przyjmują wartość OFFH. Powoduje to znaczne przyspieszenie programowania, szczególnie dla danych zawierających wiele „pułstych” bajtów.

Program „Progat” ma możliwość automatycznego rozpoznawania typu programowanego lub odczytywanego procesora. Jeżeli po umieszczeniu wybranego procesora w odpowiedniej podstawie (20 lub 40 nóżek) i uruchomieniu programu nie nastąpi automatyczne rozpoznanie procesora, to wyświetlone zostaną wszystkie możliwe typy procesorów obsługiwanych przez programator wraz z poleceniem wyboru tego właściwego, przez wciśnięcie odpowiedniego klawisza (1 do 7) na klawiaturze komputera. Przyczyną nie rozpoznania typu procesora może być:

- brak zasilania układu;
- brak procesora w podstawie;
- błędne lub niedokładne umieszczenie procesora w podstawie;
- brak sygnatury w procesorze;
- uszkodzony procesor.

Z doświadczenia wiem, że w niektórych procesorach czasami „znika” sygnatura, co uniemożliwia ich automatyczne wykrywanie. Dzieje się tak często po zbyt „brutalnym” potraktowaniu procesora, np. chwilowym odwrotnym umieszczeniu go w podstawie (odwrotne zasilanie). Jednak utrata sygnatury nie powoduje uszkodzenia procesora i można go nadal z powodzeniem wykorzystywać.

Należy pamiętać, że na raz może być programowany tylko jeden procesor w obudowie 20- albo 40-końcówkowej.

UWAGA:

Po podłączeniu układu programatora do zasilania i złącza drukarki komputera na wyjściach układów scalonych, a więc i na nóżkach podstawek pod procesory mogą występować nieokreślone stany logiczne, a na linii V_{pp} napięcie 12 V. Może to spowodować pewne komplikacje po umieszczeniu procesora w podstawie zaraz po włączeniu zasilania. Aby uniknąć przykrych niespodzianek powinno się przed pierwszym umieszczeniem procesora w podstawie uruchomić program „Progat” w celu ustalenia odpowiednich poziomów logicznych na pinach podstawek. Wystarczy uruchomić sam program „Progat.exe” bez podawania jakichkolwiek parametrów.

Wykaz elementów	
Półprzewodniki	
US1, US2,	
US3	– 74HC573 lub 74HCT573
US4	– 74LS165
US5	– CD 4040
US6	– LM 7805, patrz opis w tekście
T1, T2	– BC 547 lub podobne npn
Rezystory	
R4	– 2,2 kΩ/0,125 W
R3	– 3,9 kΩ/0,125 W
R1, R2	– 10 kΩ/0,125 W
Kondensatory	
C1, C2	– 33 pF/50 V ceramiczny
C6	– 47 nF/50 V ceramiczny
C4	– 100 nF/50 V ceramiczny
C5, C7	– 47 μF/16 V
C3	– 470 μF/16 V
Inne	
Q1	– rezonator kwarcowy od 3 do 24 MHz
PD1	– podstawka 20-, lub T24-nóżkowa
PD2	– podstawka 40-nóżkowa
płytką drukowaną numer 460	

Program „Progat” można zakupić w redakcji PE razem z płytką drukowaną programatora. W Internecie pod adresem www.pe.com.pl będzie dostępna demonstracyjna wersja programu „Progat”. Płytki drukowane wysyłane są za zaliczeniem pocztowym. Płytki i dyskietkę z programem „Progat” z dopiskiem PROGAT można zamawiać w redakcji PE.

Cena: płytką numer 460 – 11,60 zł
dyskietka PROGAT – 30,00 zł
+ koszty wysyłki.

Uwagi do „Programatora procesorów Atmel”

W nr 4/99 PE został opisany programator procesorów ATMEL. W związku z licznymi uwagami, jakie napłynęły od użytkowników wymagane jest krótkie sprostowanie. Program obsługujący programator odczytuje dane z pliku zapisanego w formacie Intel-Hex. Dane w tym pliku powinny być zapisane w kolejności wzrastających adresów, gdyż inaczej program 'progat.exe' nie będzie w stanie prawidłowo ich odczytać. Praktycznie wszystkie kompilatory programów assemblerowych generują poprawny kod w pliku Intel-Hex. Jednak kompilatory języków wyższego poziomu (np. C - KEIL) mogą wytwarzać dane w pliku, które są „pomieszane” pod względem adresów. Dlatego dla uniknięcia tego typu problemów napisałem prosty program sortujący linie z danymi w pliku Intel-Hex wg adresów. Z pewnością będzie on przydatny dla wszystkich programujących'51 w językach wyższego poziomu. Program napisany jest w Pascalu. Listing programu przedstawiam obok. Jest on także dostępny na www.pe.com.pl.

♦ **Jarosław Konieczny**

```
{ Program sortujący linie w pliku Intel-Hex wg adresow }
Program Sort_hex;
var
  licz, adr1, adr2 :word;
  t1, t2 :text;
  s, slin :string;
function Adres ( s :string ) :word;
var
  adr :word;
  kod :integer;
begin
  val ( '$' + copy(s,4,4), adr, kod );
  Adres := adr;
end;
begin
  writeln;
  writeln('Program sortujący linie w pliku Intel-Hex
  wg adresow. ');
  writeln('Sposob wywołania programu: ');
  writeln('sort_hex.exe plik1.hex plik2.hex');
  writeln('gdzie: ');
  writeln('plik1.hex - wejsciowy istniejacy plik do
  posortowania');
  writeln('plik2.hex - tworzony wynikowy plik
  posortowany');
  assign ( t1, paramstr(1) );
  reset ( t1 );
  assign ( t2, paramstr(2) );
  rewrite ( t2 );
  licz := 0;

  while not eof ( t1 ) do
  begin
    readln ( t1, s );
    if (s[1]=':') and (s[9]='0') then
      if Adres(s) = 0 then writeln ( t2, s ) else
        inc ( licz );
  end;
  adr1 := 0;
  for licz := licz downto 1 do
  begin
    adr2 := $FFFF;
    reset ( t1 );
    while not eof ( t1 ) do
    begin
      readln ( t1, s );
      if (s[1]=':') and (s[9]='0') then
        if ( Adres(s) < adr2 ) and ( Adres(s) >
        adr1 ) then
          begin
            adr2 := Adres(s);
            slin := s;
          end;
    end;
    writeln ( t2, slin );
    adr1 := adr2;
  end;
  writeln ( t2, ':00000001FF' );
  close ( t1 );
  close ( t2 );
end.
```

Zasady prenumeraty

1. Zamówienia na prenumeratę przyjmujemy począwszy od dziewiątego numeru w 1999 roku.
2. Cena dla prenumeratorów wynosi 3,80 zł za jeden egzemplarz pisma „Praktyczny elektronik” wraz z kosztami wysyłki do końca 1999 roku.

3. Gwarantujemy wysłanie wszystkich opłaconych numerów bez konieczności dopłaty w przypadku wzrostu ceny pisma w okresie objętym prenumeratą.
4. Prosimy dokonywać wpłat odpowiednio wcześniej, tak aby wypełniony kupon dotarł do Wydawnictwa w terminie do 20 września 1999 roku.

♦ **Redakcja**