

Dział "Projekty Czytelników" zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Programator uniwersalny, część 2



W drugiej, ostatniej części artykułu konstruktor programatora opisuje konstrukcje adapterów umożliwiających programowanie układów różnych rodzin oraz oprogramowanie sterujące pracą programatora.

Adaptory

Programowane układy są podłączane do programatora za pomocą adapterów. Są to proste układy zawierające podstawki pod układy scalone i kilka dodatkowych elementów dopasowujących. Takie rozwiązanie pozwoliło zmniejszyć do minimum liczbę elementów przełączających przez co zwiększono niezawodność programatora - chodzi tu szczególnie o przełączanie względnie wysokiego napięcia programującego.

Adapter dla wybranych mikrokontrolerów firm: Atmel, Microchip i pamięci EEPROM z interfejsem szeregowym

Schemat tego adaptera przedstawiono na rys. 2. Jest to prosty układ zawierający 4 podstawki pod układy scalone. Umożliwia programowanie układów ATMEL w obudowach 20-nóżkowych (zarówno MCS-51 i AVR),

PIC16x8x, pamięci EEPROM z interfejsem szeregowym typu 93Cxx oraz 24Cxx.

Adapter dla procesorów AVR firmy Atmel

Schemat adaptera przedstawiono na rys. 3. Jest to prosty układ zawierający 1 podstawkę 40-nóżkową. Umożliwia programowanie układów ATMEL AVR w obudowach z 40 wyprowadzeniami (np. AT90S4414, AT90S8515).

Adapter dla procesorów MCS-51

Schemat adaptera przedstawiono na rys. 4 (zawiera 1 podstawkę 40-nóżkową). Za jego pomocą można programować układy MCS-51 w obudowach 40-nóżkowych z pamięcią Flash (AT89C51, AT89C52, AT89C55, AT89S8252). Z powodu podawania adresu w sposób równoległy, płytkę uzupełniono o 12-bitowe liczniki 4040 (U2, U3), które spełniają funkcję licznika adresu.

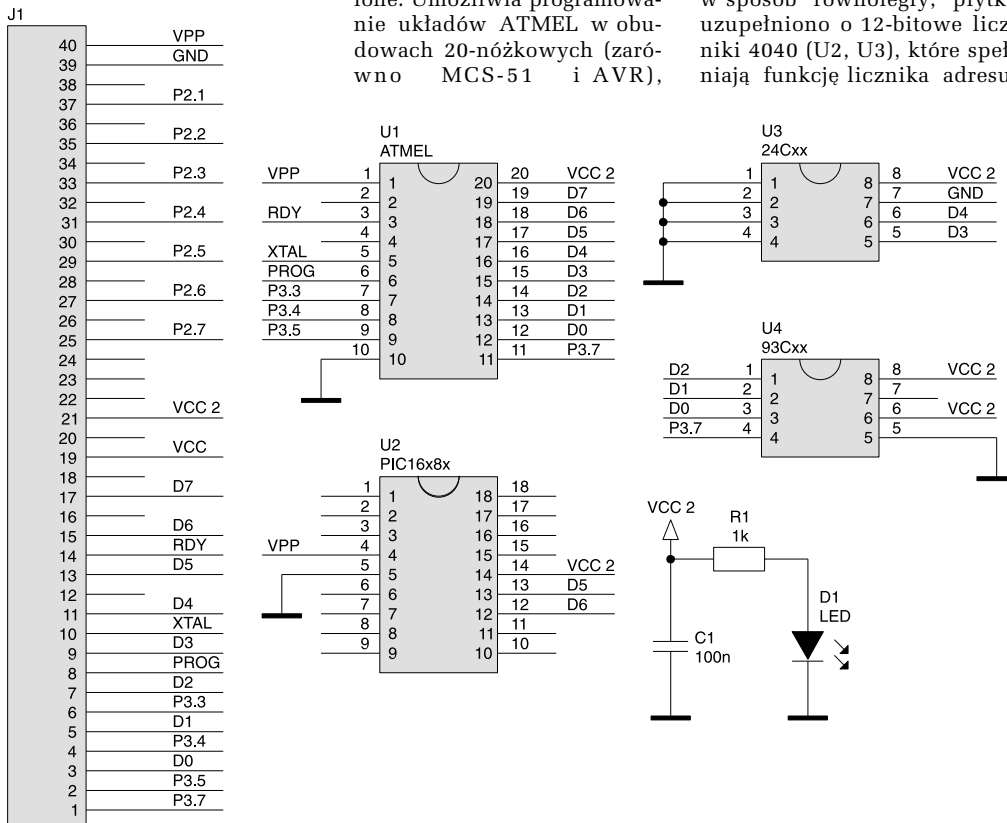
Montaż i uruchomienie

Przed rozpoczęciem montażu należy dokładnie sprawdzić wszystkie płytki drukowane, czy przypadkiem nie ma zwarcień między ścieżkami lub sprawdzić czy nie ma przerw w ścieżkach - należy to wykonać omomierzem. Nawet szerokie ścieżki, wyglądające „na oko” na sprawne, mogą być przetrawione. Po tej żmudnej, ale koniecznej czynności można przystąpić do dalszej pracy.

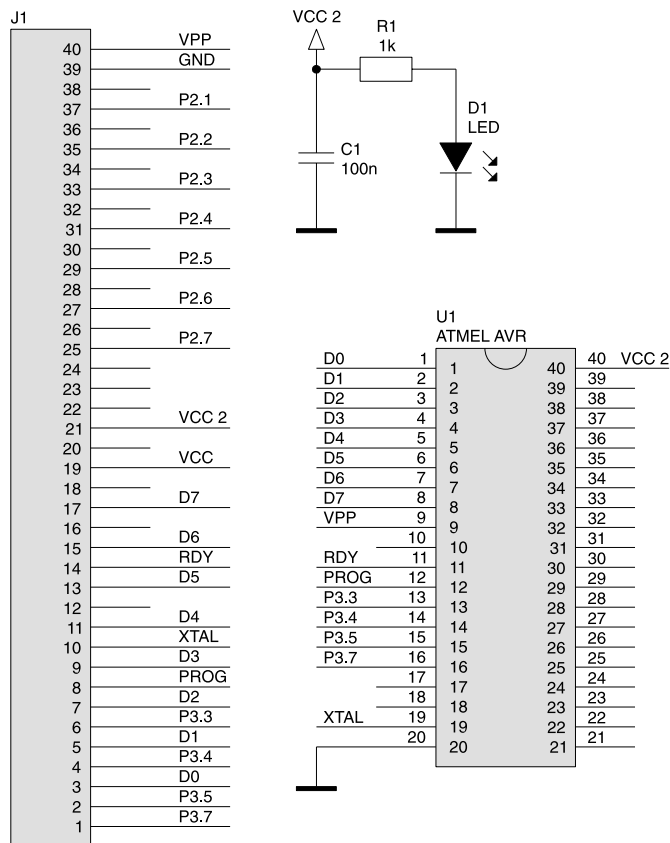
Programator należy zmontować na płytce drukowanej, w którą wcześniej należy wlotować podstawki pod układy scalone. Ułatwi to uruchamianie układu i jego późniejsze modyfikacje. Podobnie należy postąpić z płytkami adapterów.

Montaż należy rozpocząć od elementów najmniejszych gabarytowo (rezystory, kondensatory, diody, tranzystory). Następnie należy wlotować podstawki pod układy scalone, stabilizatory U1 i U2, gniazda J1, J2, J3. Połączenie głównego modułu programatora z adapterami jest wykonane za pomocą przewodu ze złączem 40-szpilkowym (tzw. „taśma” - takie samo jak w twardych dyskach IDE). Należy zwrócić uwagę na długość tego przewodu - obowiązuje generalna zasada: im krótszy przewód tym lepiej. Należy jednak zapewnić możliwość swobodnej wymiany adapterów, dlatego nie należy przesadzać ze zbytnim jego skracaniem (egzemplarz modelowy posiadał przewód długości ok. 10cm).

Po montażu następuje zawsze najciekawszy moment: uruchomienie. Przy wyjętych z podstawek ukła-



IDC40
Rys. 2.



IDC40
Rys. 3.

dach U3 i U4 podłączamy zasilanie (najlepiej o wartości ok. 26V, ale może być niższe). Powyżej 14V nie uzyskamy jedynie wyższych napięć programujących - obecna wersja jeszcze nie pozwala na programowanie układów EPROM i GAL, więc napięcie 14V w zupełności wystarcza. Dioda D3 powinna świecić. Mierzmy napięcia między wyprowadzeniami 14 i 7 układu U3 oraz 40 i 20 układu U4 - powinno wynosić ok. 5V. Sprawdzamy napięcie na wyjściu stabilizatora U1 (końcówka 3) - tu powinno być 12V. Jeśli coś jest nie tak, należy poszukać błędu. W przeciwnym przypadku wyłączamy zasilanie, wkładamy do podstawek układy U3 i U4. Podłączamy programator do dowolnego portu COM komputera PC za pomocą przewodu zakończony z obu stron wtyczkami „żeńskimi”. Należy zwrócić uwagę na „skrzyżowanie” sygnałów RXD i TXD w kablu (typowy kabel tzw. „modem zerowy”). Na komputerze PC uruchamiamy dowolny program terminalowy np. *HyperTerminal* znajdujący się w każdym pakiecie Windows 95, 98 czy NT. W terminalu

należy skonfigurować port szeregowy, do którego jest podłączony programator (COM1 do 4) na 19200 baudów, 8 bitów danych, 1 bit stopu, bez parzystości, sprzętowo sterowanie przepływem. Włączamy zasilanie progra-

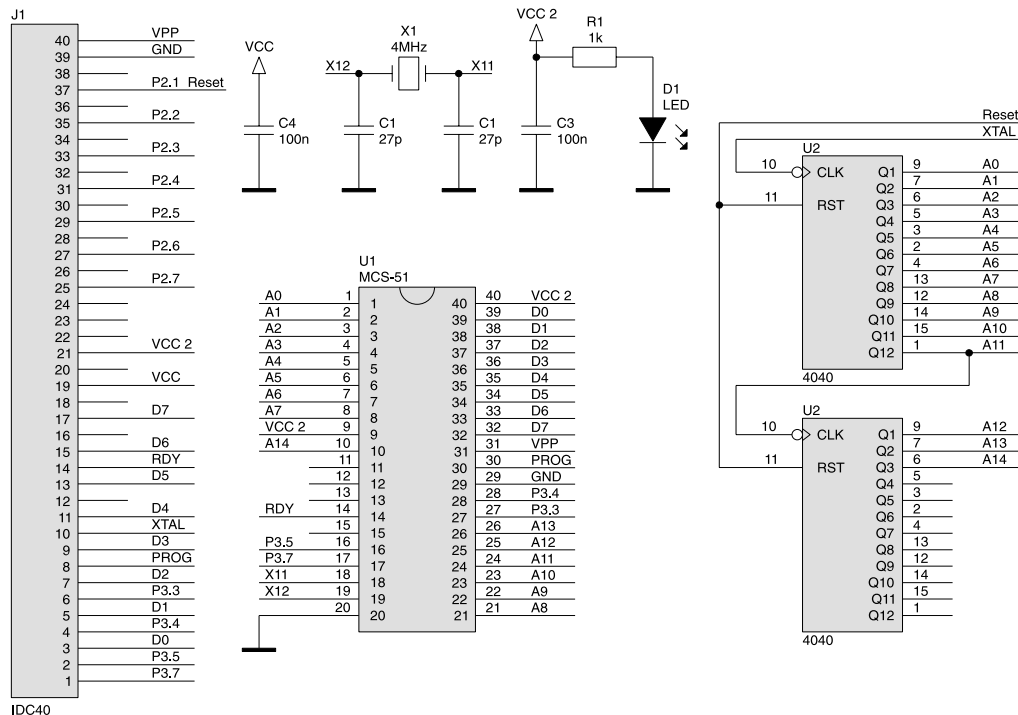
matora. Jeśli wszystko zostało dobrze skonfigurowane i podłączone, to w okienku terminala powinien się zgłosić znak zachęty następującej postaci: *RKProg>*. Naciśnięcie klawisza *ENTER* powoduje kolejne wyświetlanie znaku zachęty. Teraz przyszła pora na sprawdzenie napięć programujących: podłączamy woltomierz między masę a końcówkę 40 złącza J2, w oknie terminala wydajemy komendę *T1<ENTER>*. W odpowiedzi powinniśmy otrzymać napis 5V i takie napięcie powinien wskazywać także woltomierz. Naciśnięcie klawisza *<ENTER>* powoduje wyłączenie napięcia programującego. Wydając komendy od T2 do T5 sprawdzamy napięcia od 12 do 25V.

Jeśli powyższe czynności zostały wykonane prawidłowo i wszystko działa poprawnie, można przystąpić do próby generalnej. Zamykamy program terminalowy (żeby nie blokował portu szeregowego komputera PC). Przy wyłączonym zasilaniu podłączamy adapter układów ATMEL, PIC i EEPROM. Załączamy zasilanie programatora. Na komputerze PC uruchamiamy program *RKProg32.exe* (Win95, 98, NT - rys. 5) lub *RKProg16.exe* (Windows 3.1x). W przypadku, gdy program powita nas komunikatem o braku odpo-

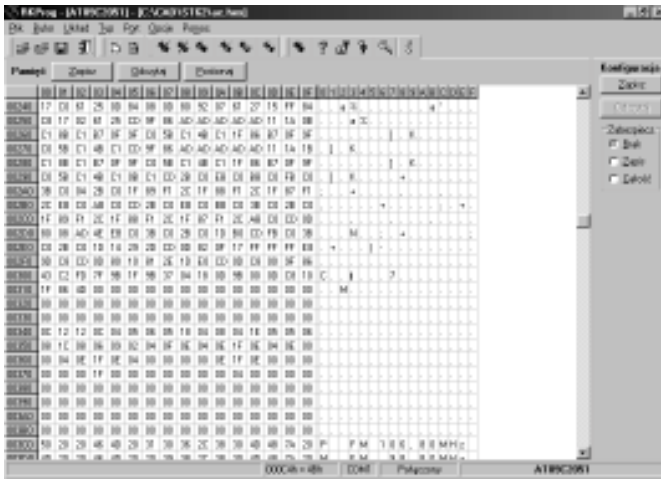
wiedzi z programatora, należy wybrać z menu *Port* pozycję odpowiadającą portowi szeregowemu, do którego jest podłączony programator. Jeżeli ponownie pojawi się ten komunikat, należy postępować zgodnie ze wskazówkami zawartymi w tym oknie. Po poprawnym zainicjowaniu komunikacji pomiędzy programem sterującym a programatorem należy umieścić w odpowiedniej podstawie programowany układ (np. pamięć 24C02) wybrać z menu *Typ* wybrany układ (tu *Typ -> EEPROM -> 24Cxx -> 24C02*). Podpowiedź, gdzie należy umieścić dany układ jest wyświetlana przez program po wybraniu polecenia z menu *Pomoc -> Położenie układu*. Następnie można przejść do odczytania zawartości układu (menu *Układ -> Czytaj*). Pozostałe polecenia obsługuje się podobnie. Teraz pozostaje tylko poznanie obsługi programu, i można wykorzystywać programator do realizacji dowolnych projektów zawierających mikrokontrolery czy pamięci nieulotne.

Zestaw komend sterujących programatorem

Dla osób chcących samodzielnie napisać program sterujący pracą programatora zamieszczamy spis wszystkich komend:



IDC40
Rys. 4.



Rys. 5.

Pierwszy znak - rodzina układu:
 'A' - układy firmy Atmel serii MCS-51 typu AT89C;
 '2' - szeregowo pamięci EEPROM z interfejsem I²C typu 24Cxx;
 '3' - szeregowo pamięci EEPROM z interfejsem Microwire typu 93Cxx;
 '7' - równoległe pamięci EPROM typu 27xx;
 'G' - układy GAL;
 'P' - układy f-my Microchip serii PIC16x8x;
 'V' - układy f-my Atmel serii AVR typu AT90Sxxxx;
 'T' - testowanie układu programatora (sprawdzanie napięć programujących).
Drugi znak - komenda do wykonania:
 'R' - odczyt pamięci;
 'W' - programowanie pamięci;
 'E' - kasowanie układu;
 'S' - odczyt sygnatury;
 'B' - test „czystości“ pamięci;
 'F' - zapis bitów konfiguracji (czasem zabezpieczenia) układu;
 'G' - odczyt bitów konfiguracji (czasem zabezpieczenia) układu;
 'O' - odczyt pamięci EEPROM niektórych procesorów;
 'P' - zapis jw.;
 'Q' - przerwanie aktualnie wykonywanej czynności.
Trzeci znak (opcjonalnie)- typ układu w ramach rodziny określanej pierwszym znakiem polecenia:
 - dla pierwszego znaku '2':
 '2' - 24C01A;
 '3' - 24C02;
 '4' - 24C04;
 '5' - 24C08;
 '6' - 24C16;
 - dla pierwszego znaku '3':
 '1' - 93C06;
 '2' - 93C46;
 '3' - 93C56;

'4' - 93C57;
 '5' - 93C66;
 - dla pierwszego znaku 'A':
 '1' - AT89C1051;
 '2' - AT89C2051;
 '3' - AT89C4051;
 '4' - AT89C51;
 '5' - AT89C52;
 - dla pierwszego znaku 'V':
 '1' - AT90S1200;
 '2' - AT90S2313;
 - dla pierwszego znaku 'P':
 '1' - PIC16F83;
 '2' - PIC16F84;
 - dla pierwszego znaku '7':
 '1' - 2716 z Upp=25V;
 '2' - 2732 z Upp=21V;
 '3' - 2732 z Upp=25V;
 '4' - 2764 z Upp=12V;
 '5' - 2764 z Upp=21V;
 '6' - 27128 z Upp=12V;
 '7' - 27128 z Upp=21V;
 '8' - 27256 z Upp=12V;
 '9' - 27256 z Upp=21V;
 'A' - 27512 z Upp=12V;
 'B' - 27010 z Upp=12V;
 'C' - 27020 z Upp=12V;
 'D' - 27040 z Upp=12V;
 'E' - 27080 z Upp=12V;
 - dla pierwszego znaku 'G':
 '1' - GAL16V8;
 '2' - GAL20V8;
 '3' - GAL18V10;
 '4' - GAL22V10.

Komunikaty synchronizujące transmisję danych, wysyłane przez programator i do programatora:
 'N' - prośba o następne dane;
 'R' - prośba o powtórzenie danych;
 'E' - błąd.

Przykład:
 Aby odczytać pamięć programu układu AT90S1200 należy wysłać komendę 'VR1'<ENTER>. W odpowiedzi otrzymamy paczkę 65 bajtów (64 bajty danych oraz bajt sumy kontrolnej). Po ich otrzymaniu należy sprawdzić

sumę kontrolną - jest ona wyliczana według następującego algorytmu zapisanego w języku C:

```
#define BL_LEN 64;
/* długość bloku danych */
unsigned char BUFFER[BL_LEN+1];
/* bufor z danymi */
suma_kontrolna=0;
/* inicjalizacja sumy kontrolnej */
unsigned char i;

for(i=0; i<BL_LEN; i++)
    suma_kontrolna+=BUFFER[i];
/* licz sumę kontrolną */
```

```
suma_kontrolna=-suma_kontrol-
na+1;
/* uzupełnienie do 2 */
```

Jeżeli wyliczona suma kontrolna zgadza się z odebraną, to można odczytać następne dane poprzez wysłanie do programatora komendy 'N'. W odpowiedzi otrzymamy kolejną porcję danych. W przeciwnym przypadku można wysłać do programatora komendę 'R' - prośbę o powtórzenie danych. Proces ten można przerwać wysyłając komendę 'Q'.

Robert Krysztof