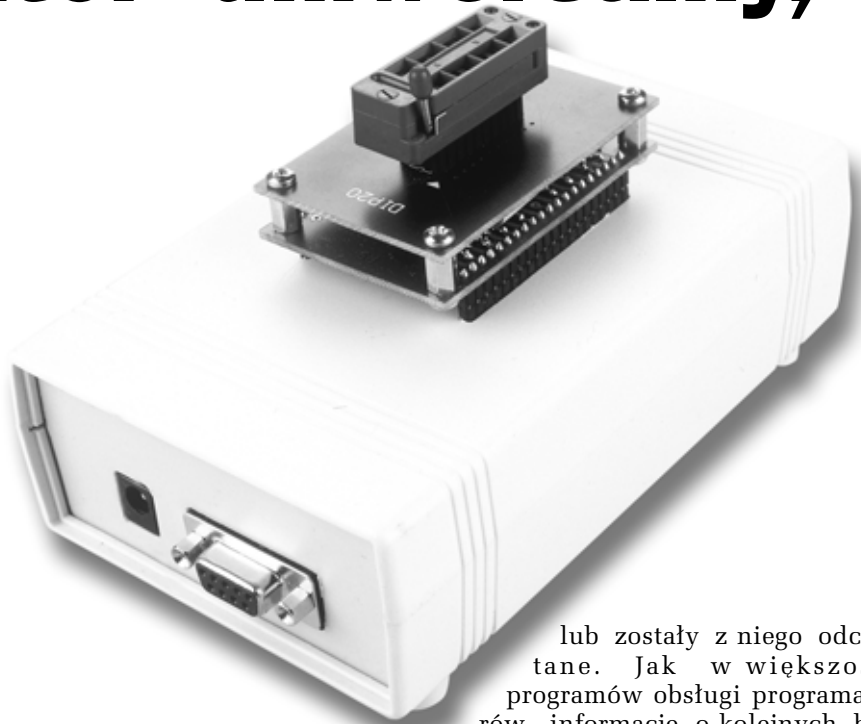


Punch Programator uniwersalny, część 2

AVT-5092

Drugą część artykułu poświęcamy przedstawieniu obsługi oprogramowania sterującego pracą programatora oraz zaleceniom, które warto uwzględnić podczas montażu urządzenia.

Rekomendacje: programator jest podstawowym przyrządem w pracowni elektronika, a więc ten opis może zainteresować większość naszych Czytelników.



lub zostały z niego odczytane. Jak w większości programów obsługi programatorów, informacje o kolejnych bajtach wyświetlane są w postaci heksadecymalnej oraz jednocześnie jako znaki ASCII. Ponad polem edycyjnym usytuowane są zwykle przyciski wyboru poszczególnych funkcji programatora i menu w trybie tekstowym, powielające funkcje przycisków. Na rys. 2 opisałem ich funkcje.

Oprócz tego, na pulpicie edycyjnym znajduje się kilka dodatkowych elementów. Na wysokości przycisków wyboru, po wczytaniu z dysku pliku danych wyświetli się jego rozmiar w bajtach. Nazwa i ścieżka dostępu do ostatnio wczytanego pliku wyświetli się na pasku statusu na dole pulpitu.

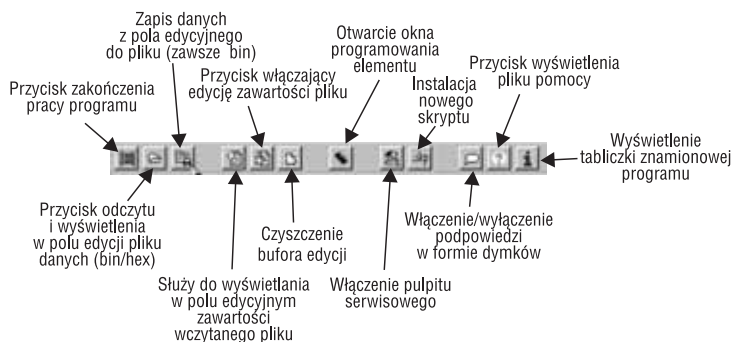
Poruszanie się po polu edycyjnym mają ułatwić elementy umieszczone po prawej stronie pulpitu (rys. 3). Są to: pasek przewijania i dwa duże klawisze. Wyświetlane standardowo pole edycyjne odpowiada rozmiarom pamięci programowanego elementu. Rozmiar zależy od typu elementu (np. w procesorze AT89C2051 można maksymalnie zaprogramować 2 kB pamięci programu) oraz od wybranej funkcji (sygnatura tego procesora to od-

Obsługa programu

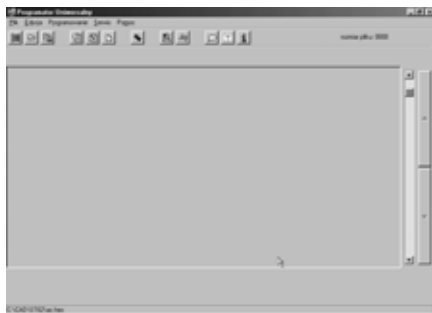
Po opisie działania programatora nadszedł czas, aby opisać jego program sterujący. Program ten jest odpowiedzialny za komfort obsługi przyrządu. O tym, jak trudno napisać program przyjazny, poprawnie działający z estetycznie wyglądającym interfejsem użytkownika miałem okazję przekonać się wielokrotnie. Program składa się z trzech wywoływanych kolejno pulpitów:

Pulpit edycji

Służą on do weryfikacji i edycji danych, które mają być zapisane w programowanym układzie



Rys. 2. Opis belki narzędziowej programu sterującego



Rys. 3. Główne okno programu z widocznymi paskami przewijania

czyt 2 bajtów). W przypadku bufora o rozmiarach do 1000h poruszanie się po nim i odnajdywanie interesujących fragmentów danych ułatwia suwak. Za pomocą dwóch dużych klawiszy można wybrać do wyświetlania kolejne segmenty danych o rozmiarze 1000h.

Pulpit wyboru funkcji programujących

Naciśnięcie przycisku otwierającego okno programowania (rys. 4) spowoduje przejście do pulpitu programowania. Za pośrednictwem tej części programu sterującego można zaprogramować element danymi znajdującymi się aktualnie w buforze edycji, odczytać do tego bufora dane z układu lub wywołać inną z dostępnych funkcji związanych z programowanym elementem np. jego kasowanie, jeżeli wyposażony jest w pamięć typu EEPROM lub Flash. Na pulpicie znajduje się sporo okienek wyświetlających informacje mające ułatwić pracę, jednak na początku mogą one trochę dezorientować operatora z powodu ich liczby. Są to głównie pola edycyjne informujące o podstawowych parametrach programowanego elementu i dostępnych opcjach. Tak jak w przypadku poprzedniego pulpitu podaję po kolei (poczynając od góry) opis i przeznaczenie każdego z jego elementów:

1. Okienko z wykazem dostępnych skryptów. Wybór skryptu poprzez dwukrotne kliknięcie nazwy spowoduje jego wczytanie. Od tego momentu programator jest gotów do programowania układów obsługiwanych przez wybrany skrypt. Wczytanie nowego skryptu spowoduje zazwyczaj zmianę zawartości pozostałych okienek znajdujących się na pulpicie.

2. W okienku z prawej strony wyświetlane są informacje, które autor skryptu przygotowanego zgodnie ze standardem języka FEMTO może w nim umieścić. Uważam, że w okienku powinna znaleźć się lista obsługiwanych przez skrypt typów elementów, informacja o producencie, typie adaptera. Ja takie informacje zamieściłem w swoich skryptach.

3. Poniżej znajduje się lista rozwijana z wykazem obsługiwanych przez skrypt elementów. W obecnej wersji oprogramowania każdy skrypt może opisywać procedury programowania do 8 elementów. Po rozwinięciu listy i kliknięciu nazwy wybranego elementu program sterujący dostosowuje swoje ustawienia do dokonanego wyboru. Ponieważ w obecnej wersji każdy skrypt opisuje elementy, których procedury programowania są identyczne, wybór oznacza w praktyce jedynie zmianę rozmiarów dostępnej do programowania pamięci. Na przykład dla procesora AT89C2051 będzie to 2 kB, natomiast w przypadku AT89C4051 rozmiar pamięci wynosi 4 kB.

4. Po wyborze elementu z listy, w sąsiednim okienku pojawią się zwięzłe informacje dotyczące wybranego elementu.

5. Poniżej linii rozdzielającej górę i dół pulpitu, z lewej strony znajduje się okienko z listą funkcji dostępnych dla wybranego układu. Dostępne funkcje to np. programowanie, weryfikacja, kasowanie.

6. Nazwa wybranej funkcji zostanie wyświetlona w okienku z prawej strony. Sposób, w jaki dokonany zostanie wybór, określają dwa przyciski znajdujące się pomiędzy okienkami.

7. Jeżeli aktywny jest przycisk z czerwoną strzałką, możliwy jest wybór pojedynczych opcji.

8. Jeżeli aktywny jest przycisk z plusem, można wybrać więcej niż jedną z dostępnych funkcji np. zapis + weryfikacja + zabezpieczenie przed odczytem. Oczywiście nie wszystkie kombinacje są sensowne, np. zapis + kasowanie + weryfikacja doprowadzi do wyświetlenia informacji o błędzie.

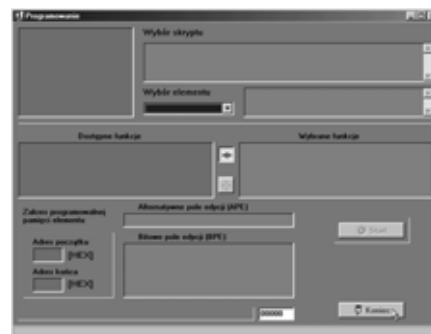
9. Jeszcze niżej, z lewej strony pulpitu znajdują się dwa okienka,

w których wyświetlany jest adres początku i końca programowanego obszaru. Jeżeli mamy do czynienia z elementem o dużej pamięci, w której zapisywanych będzie na początku niewiele danych, można dla przyśpieszenia procesu programowania ograniczyć od góry zakres programowania do rozmiaru zapisywanego bloku. Znacznie rzadziej będzie stosowane ograniczenie dolnego zakresu programowania, ponieważ np. procesory wymagają, aby pewne elementy programu rozpoczynały się od zerowego adresu komórki pamięci.

10. Obok znajdują się dwa pola edycyjne. Alternatywne pole edycji (APE, alternatywne w stosunku do pola edycji na pierwszym pulpicie) przeznaczone jest do wyświetlania małej ilości danych w formacie heksadecymalnym lub tekstowym. Może mieć zastosowanie chociażby przy wyświetlaniu odczytanej sygnatury elementu. O tym, w którym oknie informacja ta zostanie wyświetlona decyduje skrypt elementu.

11. Bitowe pole edycji (BPE) przeznaczone zostało do wyświetlania i edycji także niewielkich porcji danych w postaci bitowej (dwójkowej). Taki tryb prezentacji danych jest najwygodniejszy w przypadku odczytu lub ustawiania bitów konfiguracyjnych elementu. Jeżeli takich bitów nie ma, pole pozostaje puste.

12. Jeszcze niżej znajduje się pasek pokazujący graficznie postęp realizowanej operacji, a w okienku obok wyświetla się liczba bajtów pozostałych do zaprogramowania lub odczytania. W przypadku błędu weryfikacji, w kolorze czerwonym wyświetlony zostanie adres pierwszego bajtu różnego od bajtu w buforze edycji.



Rys. 4. Widok okna programowania



Rys. 5. Widok okna serwisowego

13. Na samym dole, na dodatkowym pasku statusu, mogą pojawiać się komunikaty, np. o bieżącej lub zakończonej operacji.

14. Naciśnięcie klawisza z prawej strony pulpitu z napisem *Start*, oznaczonego symbolem diody LED, oznacza rozpoczęcie realizacji wybranej funkcji. Spowoduje to przesłanie do części sprzętowej odpowiednich rozkazów i najczęściej dołączenie zasilania do programowanego elementu, co sygnalizowane jest świeceniem się diody LED. Przerwanie procesu programowania jest możliwe po naciśnięciu na klawiaturze komputera klawisza *escape*. Część sprzętowa przerwie programowanie i wyłączy zasilanie elementu jeszcze w jednym przypadku - jeżeli zerwana zostanie komunikacja z programem sterującym i w ciągu kilku sekund z komputera PC nie zostanie przesłany żaden rozkaz.

15. Ostatni z elementów na pulpicie - klawisz z napisem *Koniec* - zamyka pulpit funkcji programujących i otwiera pulpit z polem edycyjnym.

Pulpit funkcji serwisowych

Projektując programator, postanowiłem rozbudować możliwości jego testowania i sprawdzania poprawności działania. Ułatwi to korzystanie z programatora, szczególnie w sytuacji, gdy użytkownik przygotowuje własny adapter, a potem chciałby go sprawdzić bez używania do tego celu często kosztownego elementu programowanego. Z tego powodu funkcje serwisowe umożliwiają sterowanie i sprawdzanie stanu poszczególnych wyprowadzeń złączy JP1 i JP2.

Z lewej strony pulpitu (rys. 5) wyświetlono w uproszczeniu obydwa złącza wraz z przyjętymi dla

nich nazwami. Jeżeli przyjmujemy, że górę części sprzętowej programatora wyznacza złącze RS232, to z lewej strony znajdzie się złącze JP1, a z prawej JP2. Oprócz nazwy, obok oznaczenia każdego styku znajdzie się informacja o jego statusie: czy jest wejściem, wyjściem oraz jaki powinien być na nim poziom. W znajdujących się obok polach, po nawiązaniu łączności pomiędzy programem sterującym a częścią sprzętową programatora, wyświetlane są informacje o wersji programatora i jego numerze. Schematycznie zaznaczona zielona dioda LED zaświeci się na chwilę po każdej poprawnej transmisji między komputerem a programatorem. Czerwona dioda zaświeci się, gdy wyprowadzenie *Vcc* oznaczymy jako *ON*, co będzie oznaczać, że po wysłaniu kolejnych danych powinien załączyć się przekaźnik podający zasilanie na programowany element. Jeżeli uprzednio wczytany został skrypt konkretnego elementu, to po naciśnięciu małego przycisku oznaczonego symbolem otwartej książki zamiast standardowych nazw złączy pojawi się opis nóżki układu scalonego, do którego powinno zostać podłączone wyprowadzenie za pośrednictwem adaptera i podstawki programującej.

Z prawej strony pulpitu znajdują się narzędzia służące do wymuszania (w celach testowych) odpowiednich stanów logicznych na wybranych wyprowadzeniach złączy. W przypadku wyprowadzeń adresowych można to zrobić dwoma sposobami. Np. kliknięcie na oznaczenie *Adr5* będzie cyklicznie przełączało jego poziom z wysokiego na niski i odwrotnie. Dla wszystkich wyjść adresowych ten sam efekt można uzyskać wpisując w polu edycyjnym dwubajtową liczbę, która zostanie przekształcona na odpowiednią kombinację zer i jedynek na wyjściach adresowych. Po kolejnej transmisji danych testowych, ustawiona kombinacja zostanie przepisana na wyprowadzenia sprzętowej części programatora. W ten sposób porównując występujący stan logiczny z zaprogramowanym, można sprawdzić poprawność działania wyjść programatora.

Pozostałe wyprowadzenia, zarówno *Dat0-7*, jak i *F1-10* mogą pełnić rolę wyjść i wejść. Wyboru trybu dokonuje się zaznaczając określone pole na pulpicie. W trybie wyjścia test przebiega podobnie jak dla wyprowadzeń adresowych. W trybie wejścia można na badane złącze JP1 lub 2 podać wybrany stan logiczny, który zostanie odczytany i zasygnalizowany przy opisie wyprowadzenia. Warunkiem powodzenia tego testu jest uprzednie ustawienie badanego wejścia na poziomie wysokim.

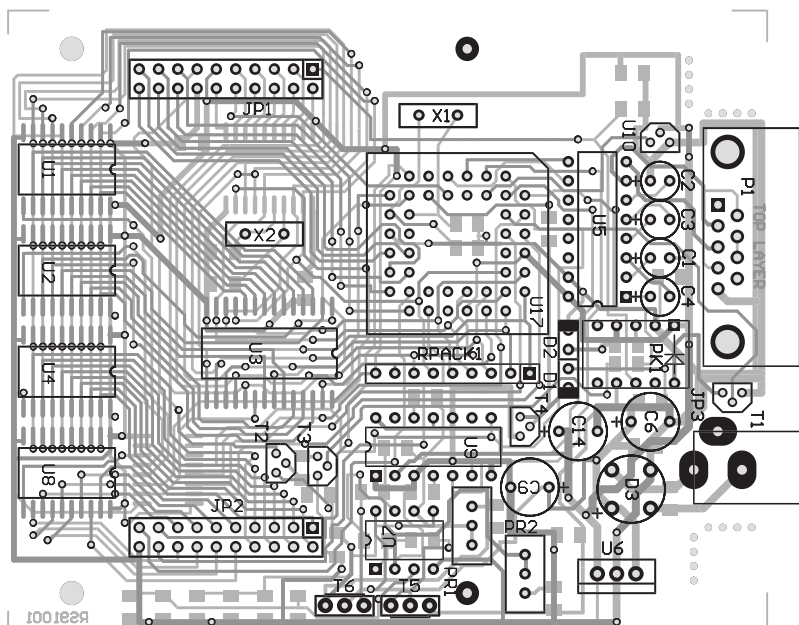
Następna sekcja na pulpicie służy do testowania wyprowadzenia napięcia zasilania *Vcc* i napięcia programowania *Vprog*. Gdy oba wyprowadzenia oznaczone są jako wyłączone, nie powinno występować na nich żadne napięcie. W czasie testu zarówno napięcia *Vcc*, jak i *Vprog* powinny być zaznaczone jako włączone, co jest sygnalizowane za pomocą symbolu świecącej się czerwonej diody LED. Użytkownik może wymusić na obu wyjściach dowolne napięcie z określonego w parametrach programatora zakresu z rozdzielczością 0,1 V. Oprócz tego, na wyprowadzeniu *F10* można wygenerować impuls o czasie trwania 1 μ s lub 1 ms.

Jeszcze niżej, za pomocą listy rozwijanej użytkownik może określić, do którego portu COM będzie przyłączona część sprzętowa programatora. Ustawienie to będzie aktualne także w czasie normalnej pracy z programatorem. Naciśnięcie znajdującego się obok klawisza *Transmisja* spowoduje jednokrotne wysłanie rozkazów z ustawionymi wcześniej parametrami wyprowadzeń. Klawisz *Koniec* zamyka funkcje serwisowe i powoduje powrót do pulpitu edycji.

Kończąc swoją pracę, program sterujący w osobnym pliku konfiguracyjnym zapisuje bieżące ustawienia. Po jego ponownym uruchomieniu będą one przywrócone, co dotyczy m.in. wybranego skryptu, wczytanego ostatnio pliku danych czy numeru portu COM, który będzie używany przez programator.

Montaż programatora

Tym czytelnikom, których zainteresował przedstawiany projekt i przystąpią do budowy programatora



Rys. 6. Rozmieszczenie elementów na płytce programatora

tora, może przydać się parę moich rad wynikających z doświadczenia, jakie zdobyłem przy budowie trzech prototypów programatora.

Programator zaprojektowany został do montażu w plastikowej obudowie oznaczonej symbolem producenta Z50. Jest ona dostępna w kilku wariantach, ja wykorzystałem pudełko o zewnętrznej wysokości 43 mm lub 36 mm. W pudełku trzeba zamontować dwie płytki drukowane. Płytką główną (dwustronna) ma wymiary 100 x 81 mm. Dla zaoszczędzenia miejsca, wszystkie oporniki i kondensatory nieelektrolityczne są przeznaczone do montażu powierzchniowego, tak jak większość układów scalonych (rys. 6). Zastosowano elementy R i C w obudowach 1206. Tym z Czytelników, którzy nie mają dobrej lutownicy o cienkim grocie i cienkiej cyny, a przede wszystkim nie mają wystarczająco dużo cierpliwości, odradzam samodzielny montaż. Źle wlutowany element SMD trudno wylutować, a w przypadku układów scalonych oznacza to najczęściej ich zniszczenie. Po tym ostrzeżeniu przystępuję do wypunktowania czynności związanych z montażem. Moim zdaniem warto przestrzegać podanej kolejności, gdyż ułatwi to pracę:

1. Płytkę drukowaną przystosowujemy do wymiarów obudowy w ten sposób, że w obu narożni-

kach (od strony gniazda DB9) wycinamy kwadraty o wymiarach mniej więcej 10 mm (zaznaczono na płytce), aby zrobić miejsce na wewnątrz prowadzone wkręty mocujące górę i dół obudowy.

2. Na dolnej stronie płytki lutujemy wszystkie oporniki i kondensatory SMD. Można najpierw zwilżyć cyną jeden z punktów lutowniczych i przylutować końcówkę elementu. Po sprawdzeniu, że przylega on do płytki i ewentualnej korekcie jego położenia, lutujemy jego drugą końcówkę. Radzę się nie śpieszyć, co zmniejszy ryzyko pomyłki. O ile bowiem oporniki mają wytrawione na powierzchni cyfry z oznaczeniem oporności, to kondensatory pozbawione są takich oznaczeń i mogą nie różnić się wyglądem przy zupełnie różnej pojemności.

3. Od dołu wlutowujemy jeszcze układ multipleksera U12 i chwilowo przestajemy się interesować dolną częścią płytki.

4. Na górze należy wlutować montowane powierzchniowo układy scalone U1, U2, U4, U8, U3. Tak jak w przypadku elementów RC, najpierw należy przylutować jedną, najlepiej skrajną nóżkę układu, sprawdzić jego położenie i dopiero potem lutować pozostałe nóżki.

5. Wlutowujemy pozostałe elementy od najmniejszych do największych. Lutowanie małego tranzystora, gdy wokoło znajdują

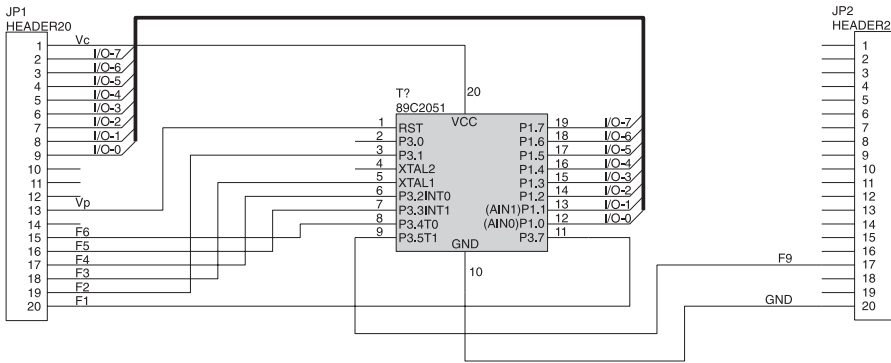
się wysokie elementy, może być bardzo stresujące. Proszę zwrócić uwagę na prawidłową lokalizację markera podstawki PLCC procesora. Ścięta krawędź powinna znaleźć się u góry po lewej stronie. Kwarc X2, jeżeli nie jest w niskiej obudowie, radzę położyć, co zapobiegnie jego wyłamaniu przez kable. Przed jego wlutowaniem, obszar płytki drukowanej, do której może dotykać metalowa obudowa kwarcu, należy zabezpieczyć przez przyklejenie np. kawałka taśmy izolacyjnej. Jako ostatnie należy wlutować wysokie elektrolity i gniazdo DB9.

6. Dla poprawy warunków pracy stabilizatora U6 warto zwiększyć jego chłodzenie, stosując mały radiator.

7. Jako złącza JP1 i JP2 można zastosować dwurzędowe, kwadratowe szpilki stosowane w złączach komputerowych. Podobnych można użyć jako złącza dla diod LED. Zastosowanie złącz zamiast lutowanych na stałe przewodów ułatwi zarówno montaż, jak i demontaż programatora, co może się przydać szczególnie w fazie uruchamiania układu.

8. Następnie należy wykonać czynności mechaniczne związane z przystosowaniem obudowy do potrzeb programatora. W tym celu od strony gniazda RS-a i gniazda zasilającego w plastikowej płytce obudowy należy wyciąć prostokątne otwory na te gniazda.

9. Druga płytka programatora jest płytką jednostronną. Przewidziano na niej miejsce na dwa gniazda wyprowadzające na zewnątrz obudowy sygnały złącza JP1, JP2. W czasie normalnej pracy, do złączy będą wkładane adaptory programujące. Ponieważ gniazda te są przewidziane do intensywnej pracy (częste wkładanie i wyjmowanie adapterów), a jednocześnie muszą zapewniać pewny styk, muszą być dobrej jakości. Najodpowiedniejsze wydają się listwy tzw. gniazd precyzyjnych o średnicy ok. 0,8 mm (średnica większa niż gniazd precyzyjnych dla układów scalonych), których wewnętrzne styki pokryte są stopem z domieszką złota. Gniazda powinny być wlutowane dokładnie prostopadle do płaszczyzny płytki. Dodatkowo na płytce przewidziano miejsce na opor-



Rys. 7. Schemat elektryczny adaptera dla mikrokontrolerów 89C2051/4051

niki podciągające poziom niektórych sygnałów do napięcia +5V, co może być konieczne przy długich przewodach łączących płytę główną programatora z płytą gniazd.

10. W górnej części obudowy należy wyciąć dwie równoległe szczeliny na opisywane wyżej gniazda. Dodatkowo trzeba wywiercić dwa otwory na diody LED o średnicy zależnej od wymiarów zastosowanych elementów i ich oprawek. Moim zdaniem najlepiej mocuje się diody o średnicy 5 mm. Otwory powinny znaleźć się na obudowie od strony operatora, aby w czasie pracy programatora diody były widoczne.

11. Do zamocowania płytki gniazd do górnej części obudowy (tej z dwiema wyciętymi wcześniej szczelinami) użyłem 4 wkrętów M3 o długości ok. 10 mm z łbami stożkowymi i kleju POXIPOL. Wkręty mocujące płytkę powinny być przyklejone do górnej części obudowy po jej wewnętrznej stronie. Miejsca klejenia należy koniecznie zmatowić albo zarysować, gdyż w przeciwnym przypadku żywica nie skleji wkrętów z plastikiem dostatecznie mocno. Klej powinien oblewać dolną część główki wkrętu, jednak nie może sięgać do części nagwintowanej. Zanim klej stężeje, korzystamy z płytki złącz i kory-

gujemy położenie klejonych wkrętów tak, aby ich położenie pokrywało się z otworami w płytce. Oczywiście, trzeba to robić ostrożnie, żeby nie przykleić płytki do obudowy. Klej typu POXIPOL twardnieje już po 10 minutach. Przed upływem tego czasu wyjmujemy płytkę, mając pewność, że wkręty znajdują się na właściwym miejscu, natomiast płytka nie skleji się z obudową.

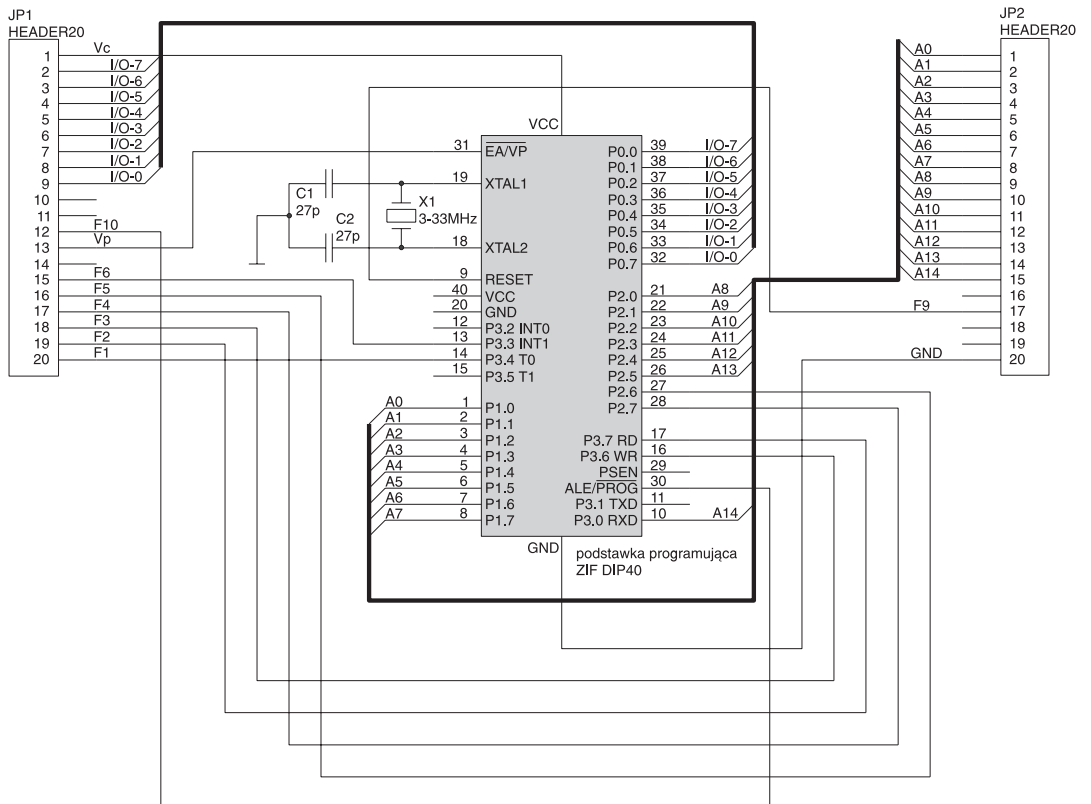
12. Do płytki złącz należy teraz dolutować przewody, które ją łączą ze stykami JP1 i JP2 płytki głównej. Takie połączenie można wykonać za pomocą dwóch odcinków 20-żyłowej taśmy o długości 8...10 cm zakończonych gniazdami zaciskowymi. Po roz-

dzieleniu z obu stron taśmy żył na długości 15...20 mm i usunięciu izolacji, końcówki należy poobielić cyną. Taka taśma nie lutuje się najlepiej i wskazane jest użycie kalafonii (sam topnik zawarty w lucie może nie wystarczyć). Taśmę należy ułożyć równoległe na płytce i dolutować do gniazd żyła po żyłę, zaczynając od skrajnych i skracając w miarę konieczności środkowe. Takie ułożenie taśm połączeniowych, które po skręceniu obudowy ułożą się w jej wnętrzu „esowato“, zapobiegnie ich ruchom w miejscach lutowania ze złączem i w efekcie obrywaniu przewodów.

13. Ostatnią czynnością tej części montażu jest przykręcenie płytki złącz do górnej części obudowy za pomocą przyklejonych wkrętów oraz zamocowanie LED-ów z przewodami i wtykiem.

Budowa adapterów

Zadaniem wymiennych adapterów jest doprowadzenie sygnałów ze złącz JP1 i JP2 do podstawki programującej typu ZIF, a za jej pośrednictwem do odpowiednich wyprowadzeń programowanego elementu. Każdy adapter składa się z dwóch jednostronnych płytek drukowanych.



Rys. 8. Schemat elektryczny adaptera dla mikrokontrolerów 89C51/52

Dolna, w projekcie określona nazwą *adapter base*, jest taka sama dla wszystkich typów adapterów. Służy do wlutowania dwóch rzędów styków, które w momencie mocowania adaptera znajdują się w gniazdach JP1 i JP2. Utworzone w ten sposób złącze powinno zapewniać pewny styk, a jednocześnie pozwalać na rozłączenie bez używania nadmiernej siły. Najlepiej do tego celu nadają się listwy styków tzw. złoconych okrągłych o średnicy 0,8 mm. Gorsze są styki o przekroju kwadratowym. Zdecydowanie odradzam używanie styków pokrytych stopem w kolorze srebrnym. Tworzone połączenie jest co prawda niezawodne, ale wyjęcie adaptera z takimi stykami z gniazda wymaga dużej siły i może grozić uszkodzeniem złącz.

Druga (górną) płytką adaptera ma otwory mocujące przystosowane do użytej podstawki ZIF. Podstawki mogą się od siebie różnić tak, jak różnią się typy obudów programowanych elementów np.

DIP, PLCC, FPGA itd. Różna może być także liczba wyprowadzeń, chociaż z reguły podstawki większej np. DIP40 można użyć do programowania elementów DIP o 20 nogach. Podstawki można albo na stałe wlutować do górnej płytki adaptera, albo osadzać je za pośrednictwem złączy wykonanych ze styków precyzyjnych. Tego typu rozwiązanie nie jest estetyczne, ale umożliwia wykonywanie jednej podstawki ZIF (zwykle stosunkowo drogiej) w kilku adapterach.

Obie płytki są ze sobą łączone przewodami w sposób przypisany przez skrypt, z którym współpracują. Struktura pliku skryptu pozwala na umieszczenie w nim takich informacji przez programistę. Potem mogą być one wyświetlone przez program sterujący programatorem na pulpicie funkcji serwisowych po naciśnięciu przycisku z ikoną książki. Przykładowy schemat połączeń opisany w moich skryptach opracowanych dla programowania proceso-

rów AT89C2051/4051 i AT89C51/52 przedstawiono na **rys. 7 i 8**.

Po wykonaniu połączeń obie płytki należy ze sobą złączyć, tworząc coś w rodzaju kanapki, z przewodami pomiędzy płytkami. Do mocowania można użyć 4 odcinków grubej srebrzanki i przylutować je od strony druku obydwu płytek. Można także użyć gwintowanych wewnętrznie tulejek dystansowych zamontowanych w czterech rogach pomiędzy obydwoma płytkami. Tulejki powinny mieć długość 6 - 8 mm. Wkręty użyte do takiego montażu powinny być oczywiście krótsze, o długości gwintu maksymalnie 5 mm.

Ryszard Szymaniak, AVT
ryszard.szymaniak@ep.com.pl

Opis języka i ewentualnych zmian jest dostępny na stronie <http://www.aries-rs.com.pl/femto>.

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/styczen03.htm>.