

Uniwersalny programator mikrokontrolerów PIC, część 2

AVT-573

Stwierdzenie, że mikrokontrolery PIC zdominowały nasz rynek, byłoby chyba lekko przesadzone, faktem jednak jest, że są one dość lubiane przez naszych elektroników-konstruktorów. Wymaganiem w dzisiejszych czasach standardem jest posiadanie wewnętrznej pamięci programu, najlepiej programowanej w układzie. PIC-e warunek ten spełniają. Trzeba tylko wiedzieć, jak tę pamięć zaprogramować i oczywiście mieć czym to zrobić.

Rekomendacje: Przedstawiony w artykule układ jest uniwersalnym systemem umożliwiającym programowanie mikrokontrolerów firmy Microchip.

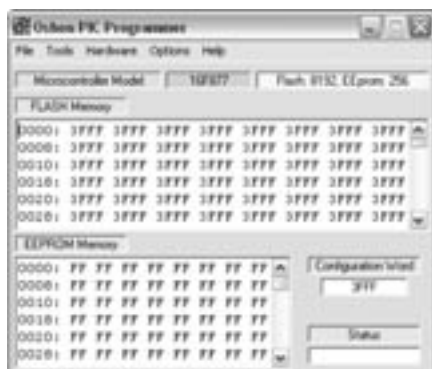
Oprogramowanie

1. Programator „Oshon PIC Programmer”

Program jest dostępny na stronie autora www.oshonsoft.com. Pobrany plik należy rozpakować i zainicjować instalację. Po zainstalowaniu programator jest gotowy do pracy. Ważną cechą jest fakt, że program posiada zaimplementowane sterowniki umożliwiające pracę w systemie Windows 98, jak również Windows NT oraz XP. Okno programatora jest przedstawione na **rys. 7**. W oknie tym widoczna jest informacja o typie programowanego mikrokontrolera, wielkości pamięci programu oraz pamięci danych EEPROM. Dodatkowo przestawiona jest zawartość tych pamięci. Programator posiada bardzo prosty interfejs i przeznaczony jest do programowania rodzin układów: PIC16F62x, PIC16F7x, PIC16F8x oraz PIC16F87x.

Funkcje paska wyboru

File – w tym menu znajduje się polecenie *Clear buffer* oraz *Open*. *Clear buffer* służy do kasowania buforów programatora dla pamięci Flash oraz EEPROM. Do komórek pamięci Flash zostaje wpisana wartość 0x3FFF, a do komórek pamięci EEPROM wartość 0xFF. Polecenie *Open* umożliwia wyczytanie do bufora programato-



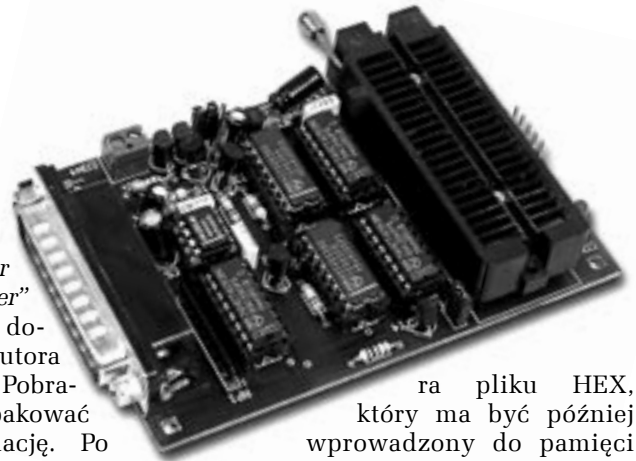
Rys. 7. Wygląd okna Oshon PIC programmer



Rys. 8. Okno konfiguracji linii portu LPT



Rys. 9. Okno testowania programatora

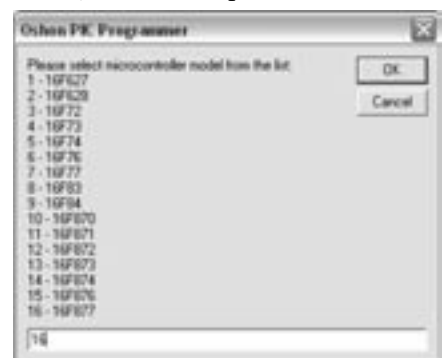


ra pliku HEX, który ma być później wprowadzony do pamięci mikrokontrolera.

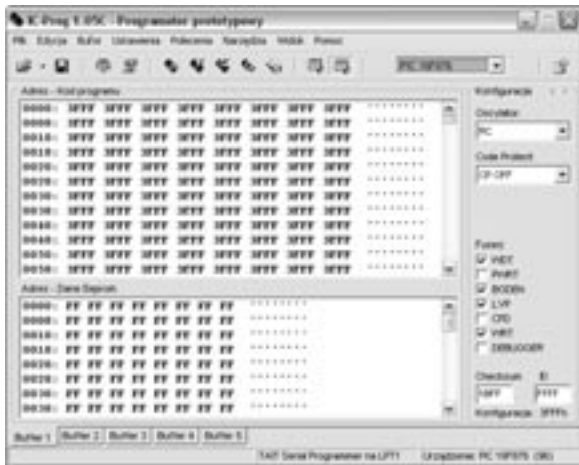
Tools – w tym menu zawarte są polecenia zapisu, odczytu i weryfikacji pamięci.

Hardware – menu to dotyczy parametrów dołączonego programatora. Opcja *Schematics* wyświetla okno ze schematem prostego programatora współpracującego z oprogramowaniem. W zależności od zastosowanego układu pełniącego rolę buforów pomiędzy złączem komputera a mikrokontrolerem należy odpowiednio skonfigurować poszczególne linie portu LPT. W przedstawionym programatorze bufora te powodują zanegowanie wszystkich linii sterujących, dlatego należy skonfigurować oprogramowanie tak, aby ponownie zanegowało te sygnały, co w efekcie pozwoli na poprawną współpracę programatora z oprogramowaniem.

Hardware Settings – służy do konfiguracji wyjść (**rys. 8**). Możliwe jest także sprawdzenie dzia-



Rys. 10. Okno wyboru typu mikrokontrolera

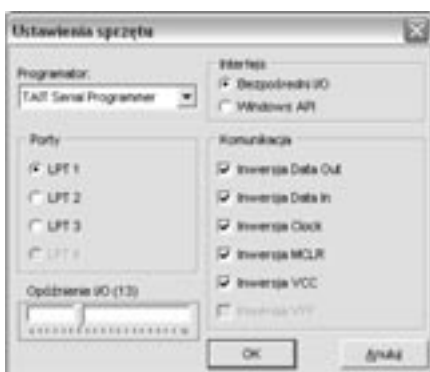


Rys. 11. Wygląd okna programu ICProg

łania dołączonego programatora za pomocą opcji *Check Hardware* (rys. 9), gdzie można zmienić stan poszczególnych linii sterujących i wykryć ewentualne nieprawidłowości w pracy programatora.

Options – w tym menu dokonuje się wyboru obsługiwanego mikrokontrolera. Po wybraniu jedynej opcji *Change Microcontroller Model* ukaże się okno przedstawione na rys. 10. Aby wybrać konkretny model mikrokontrolera, należy wpisać cyfry przypisane do typu, a następnie zatwierdzić je.

Proces programowania przebiega następująco: W menu *Options->Change Microcontroller Model* należy wybrać odpowiedni model mikrokontrolera, następnie wczytać plik HEX i wybrać polecenie *Tools->Programm All*. Należy zwrócić uwagę na fakt, że wczytany plik musi zawierać słowo konfiguracyjne (wartości tzw. bezpieczników), gdyż z poziomu programatora nie można ustawić tego parametru. W czasie programowania widoczny jest licznik programowanych słów pamięci programu i danych. Po zaprogramowaniu



Rys. 12. Okno wyboru typu programatora.

należy wydać komendę weryfikacji kodu *Tools->Verify*. Jeśli pamięć zostanie zapisana błędnie, to pojawi się komunikat o adresie pierwszego błędnego słowa, w przeciwnym przypadku programowanie należy uznać za zakończone.

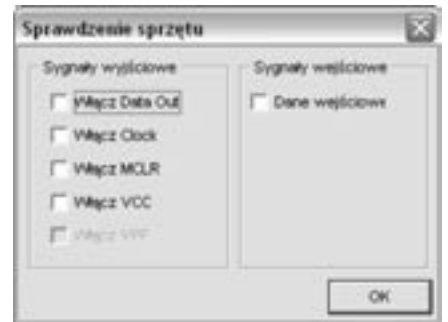
Programator IC-Prog

Programator IC-Prog jest kolejnym darmowym narzędziem umożliwiającym programowanie mikrokontrolerów.

Oprogramowanie jest udostępniane na stronie www.icprog.com. Tam także znajdują się uaktualnienia oraz schematy obsługiwanych programatorów. Ich liczba jest duża, gdyż oprogramowanie umożliwia programowanie niemalże wszystkich typów mikrokontrolerów, nie tylko firmy Microchip, ale również Atmel. Możliwe jest także programowanie szeregowych pamięci EEPROM. Wygląd okna programu jest przedstawiony na rys. 11.

Najcenniejszą cechą tego oprogramowania jest możliwość wybrania jednej z wielu wersji językowych, w tym także polskiej. Kolejną zaletą jest brak konieczności instalowania programu, gdyż plik wykonywalny od razu uruchamia programator. Oprogramowanie może pracować w systemach Windows 98 lub niższym, a także w systemach Windows NT oraz XP. Do pracy w systemach ostatnio wymienionych wymagane jest dołączenie biblioteki umożliwiającej dostęp do portów komputera. Biblioteka ta o nazwie *icprog.sys* jest również dostępna na stronie autora programu.

Przy pierwszym uruchomieniu programu należy skonfigurować parametry jego pracy. W celu ułatwienia wprowadzania dalszych ustawień, na wstępie należy ustawić polski język wyświetlanych komunikatów. Wykonuje się to w menu *Settings->Options->Language*, w wyświetlonym oknie należy wybrać opcję *Polish*. Po tej czynności program samoczynnie zostanie zamknięty i ponownie uruchomiony z interfejsem w języku polskim. Następnie na-

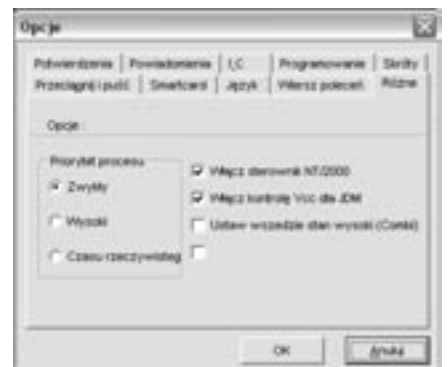


Rys. 13. Okno testowania dołączonego programatora

leży wybrać rodzaj odpowiedniego programatora w menu *Ustawienia->Sprzęt* lub nacisnąć klawisz funkcyjny F3. W otwartym oknie (rys. 12) należy wybrać typ programatora – *TAIT Serial Programmer* oraz port równoległy, do którego jest dołączony. W oknie tym można także ustawić opóźnienie wysyłania danych oraz zanegować poszczególne linie danych. Dla przedstawionego programatora wszystkie linie należy ustawić według rys. 12.

Oprogramowanie umożliwia także sprawdzenie poprawności pracy dołączonego programatora. Funkcja ta jest wywoływana w menu *Ustawienia->Sprawdź sprzęt*. Widok uruchomionego okna jest przedstawiony na rys. 13. Testowanie przebiega analogicznie jak to miało miejsce w przypadku programu Oshon Programmer.

Jeżeli programator będzie pracował w systemie Windows XP, to należy do katalogu, w którym znajduje się *ICProg* skopiować plik *icprog.sys* oraz włączyć sterownik systemu NT. Dokonuje się tego poprzez menu *Ustawienia->Opcje*. W nowo otwartym oknie należy w menu *Różne* zaznaczyć opcję *Włącz sterownik NT/2000* (rys. 14).



Rys. 14. Okno dodatkowych opcji programowania

UWAGA! Przeprowadzone testy wykazały, że przy szybkim procesorze komputera mogą pojawiać się błędy w czasie programowania. Częściowo można je wyeliminować odpowiednim ustawieniem opóźnienia portu (rys. 12). Dobre efekty przynosi również uruchomienie innej aplikacji obciążającej procesor. Dla komputera z procesorem 850 MHz, uruchomienie odtwarzacza MP3 powoduje całkowity brak błędów. Wymienione problemy występują dla programu IC-Prog w wersji 1.05C, w przypadku wersji 1.04 programowanie przebiega prawidłowo bez dodatkowych zabiegów. Jednak ta wersja programu obsługuje mniejszą liczbę mikrokontrolerów. Dlatego w zależności od programowanego mikrokontrolera należy zastosować odpowiednią „metodę”.

W menu *Programowanie* można wybrać sposób weryfikacji zapisywanego kodu w pamięci mikrokontrolera. Do wyboru jest: weryfikacja w czasie programowania lub po zaprogramowaniu. W menu *Skróty* można zapisać cztery najczęściej używane typy mikrokontrolerów, następnie zaprogramowany typ będzie można wybrać za pomocą kombinacji klawiszy CTRL+F1/F2/F3/F4.

Do operacji na pamięci mikrokontrolera służą polecenia zawarte w menu *Polecenia*:

- *Odczytaj wszystko* (F8) – odczytuje zawartość pamięci Flash oraz EEPROM mikrokontrolera i umieszcza w buforze, skąd następnie dane mogą zostać zapisane w postaci pliku HEX lub BIN.
- Polecenie *Programuj wszystko* (F5) – zapisuje zawartość bufora do pamięci Flash i EEPROM mikrokontrolera.
- *Programuj i konfiguruj* (F4) – zapisuje do mikrokontrolera konfigurację bezpieczników.
- *Wyczyść wszystko* – kasuje pamięć programu i danych mikrokontrolera.
- *Weryfikuj* – porównuje dane zapisane w pamięci mikrokontrolera z danymi znajdującymi się w buforze.

Procedura programowania wygląda następująco: na początku należy wybrać typ mikrokontrolera z listy lub poprzez kombinacje

klawiszy CTRL+F1/F2/F3/F4 (jeżeli wcześniej zostały przypisane do nich odpowiednie typy mikrokontrolerów), następnie należy wczytać odpowiedni plik kombinacją klawiszy CTRL+O (litera „O”) lub poleceniem *Otwórz...* Jeżeli wczytany plik zawiera prawidłowe słowo konfiguracyjne, to automatycznie zostaną ustawione odpowiednie bezpieczniki (*Fuses*) w oknie dialogowym programatora, w przeciwnym wypadku ustawienia te można zmienić ręcznie. Proces programowania rozpoczyna się naciśnięciem klawisza F5 lub wybraniem polecenia *Programuj wszystko*. W zależności od wybranej opcji weryfikacji, poprawność zapisu jest sprawdzana podczas programowania lub po jego zakończeniu. Jeżeli nie pojawi się komunikat o błędzie, to pamięć mikrokontrolera została zaprogramowana prawidłowo.

Oprócz programowania okienkowego możliwe jest także wywołanie procesu programowania przez zewnętrzny program z linii poleceń. Funkcja ta jest przydatna podczas pisania programu w edytorze, na przykład MPLAB. Wtedy z poziomu tego edytora można wywołać proces programowania. Ułatwia to znacznie pracę przy wielokrotnym sprawdzaniu poprawności skompilowanego programu. Wywołanie programu z linii poleceń ma następującą postać: *Icprog polecenie1 polecenie 2 po-*

leciecie 3... Jako polecenia stosowane są znaki, których znaczenie jest następujące:

p – programuj układ,
v – sprawdź poprawność zapisu,
r – odczytaj pamięć układu,
i – przy programowaniu nie pokazuj okna z zawartością pamięci, pokazywane jest tylko okno informacyjne wskazujące stopień zaawansowania wykonywanego procesu,

l – wczytaj plik,
fxxxh – słowo konfiguracyjne (dla bezpieczników),

q – zamknij okno po zaprogramowaniu.

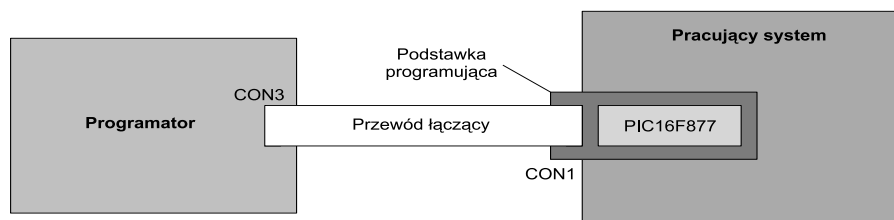
Przykładowa komenda programowania może być następująca:

IC-Prog -ltest.hex -p -v -i -f3E-EA -q

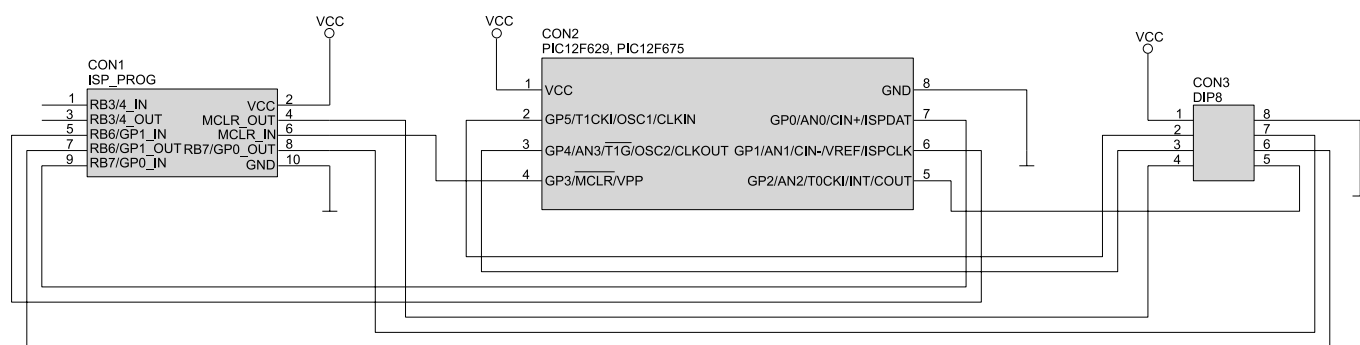
Po wydaniu takiej komendy zostanie uruchomiony IC-Prog, następnie będzie wczytany plik test.hex, którym zostanie zaprogramowany mikrokontroler, dalej zostanie zweryfikowana poprawność zapisu i zapisane słowo konfiguracyjne. Okno programatora będzie wyświetlone w postaci zminimalizowanej (tylko wskaźnik programowania), a po zakończeniu procesu programowania zostanie automatycznie zamknięte. Jak wi-

Tab. 3. Zestawienie mikrokontrolerów obsługiwanych przez odpowiednie podstawki programujące

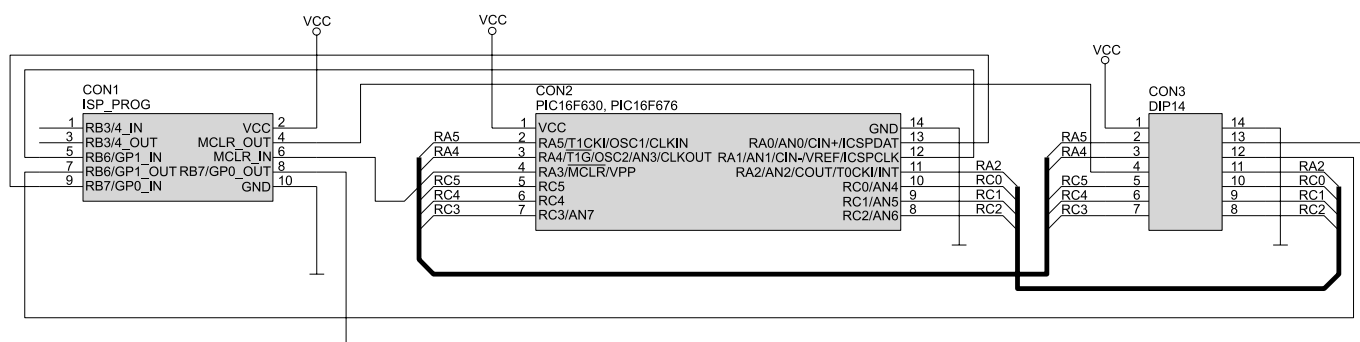
Podstawka	Obsługiwane układy
DIP8	PIC12F629, PIC12F675
DIP14	PIC16F630, PIC16F676
DIP18-1	PIC16F83, PIC16F84, PIC16F84A, PIC16F627, PIC16F628
DIP18-2	PIC16F83, PIC16F84, PIC16F84A, PIC16F818, PIC16F819
DIP18-3	PIC16F83, PIC16F84, PIC16F84A, PIC18F1320
DIP28-1	PIC16F72, PIC16F73, PIC16F76, PIC16F870, PIC16F872, PIC16F873, PIC16F873A, PIC16F876, PIC16F876A
DIP28-2	PIC18F242, PIC18F252, PIC18F248, PIC18F258, PIC18F2320,
DIP40-1	PIC16F74, PIC16F87, PIC16F871, PIC16F874, PIC16F874A, PIC16F877, PIC16F877A
DIP40-2	PIC18F442, PIC18F252, PIC18F448, PIC18F458, PIC18F4320, PIC18F4539



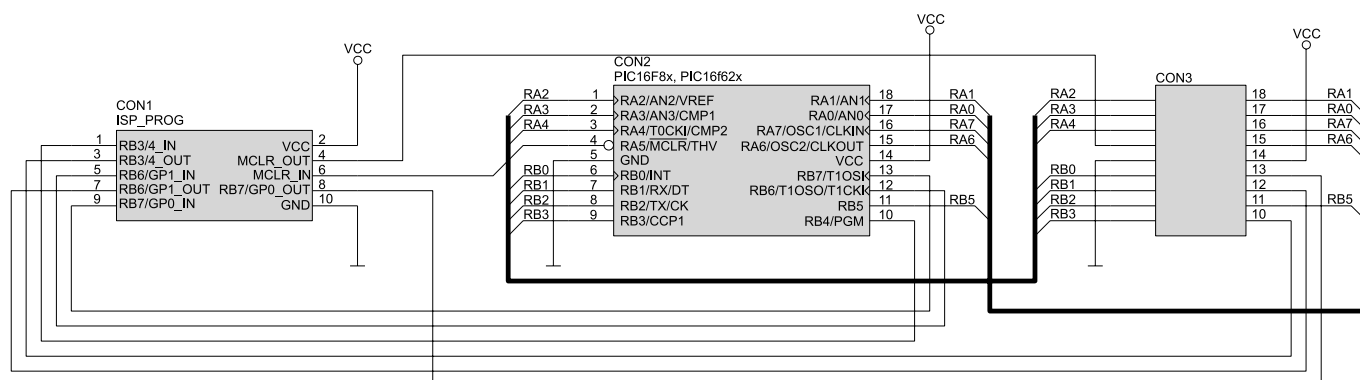
Rys. 15. Sposób podłączenia programatora do pracującego systemu



Rys. 16. Schemat elektryczny adaptera DIP8



Rys. 17. Schemat elektryczny adaptera DIP14



Rys. 18a. Schemat elektryczny adaptera DIP18-1

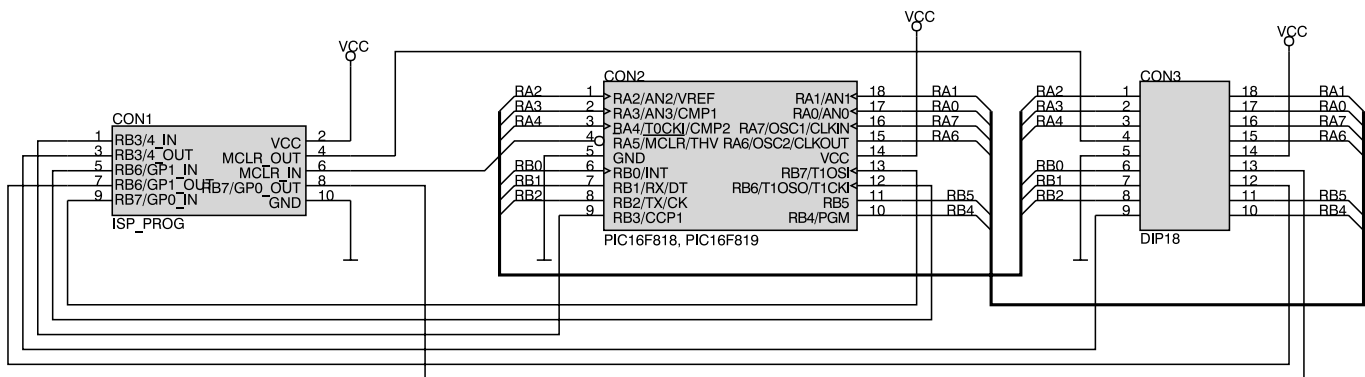
dać, taki tryb programowania jest bardzo przydatny w przypadku prac nad projektem, natomiast przy programowaniu plikami wynikowymi wygodniejszy będzie standardowy sposób programowania za pomocą klawiszy funkcyjnych lub myszki.

Akcesoria dodatkowe

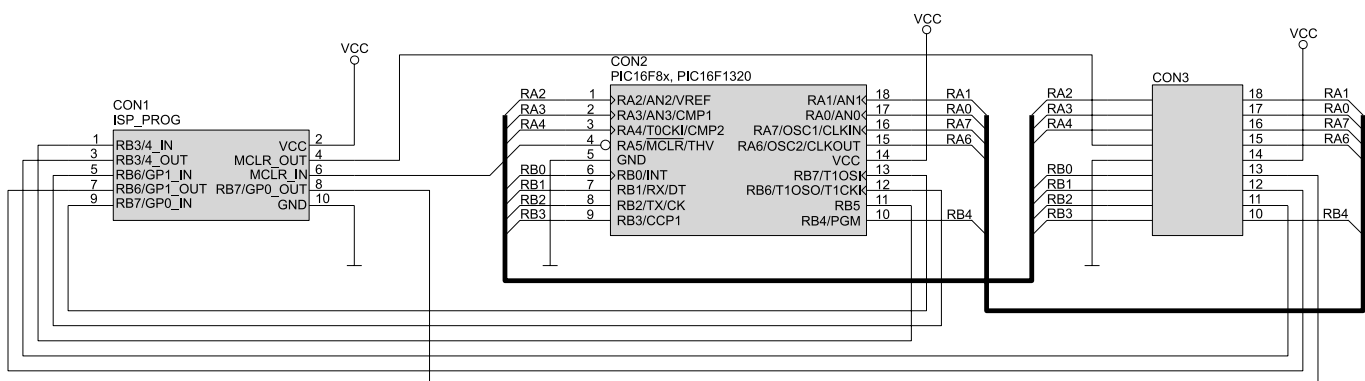
Do programowania w systemie należy zastosować dodatkowe podstawki programujące, które przyłączane są do programatora poprzez złącze CON3. W zależności od typu mikrokontrolera należy zastosować odpowiednią podstawkę. Ze względu na podobieństwa pew-

nych grup mikrokontrolerów, jedna podstawka może być zastosowana do kilku rodzajów układów. Każda podstawka posiada złącze sygnałów wejściowych CON1, które należy połączyć przewodem taśmowym ze złączem CON3 płytki programatora. Na każdej podstawie programującej znajduje się jedna podstawka służąca do zamontowania mikrokontrolera oraz druga umożliwiająca połączenie z podstawką systemu. Sposób podłączenia programatora z pracującym systemem jest przedstawiony na rys. 15. Do obsługi mikrokontrolerów wymienionych w tab. 2 (w cz. 1 artykułu) niezbędne

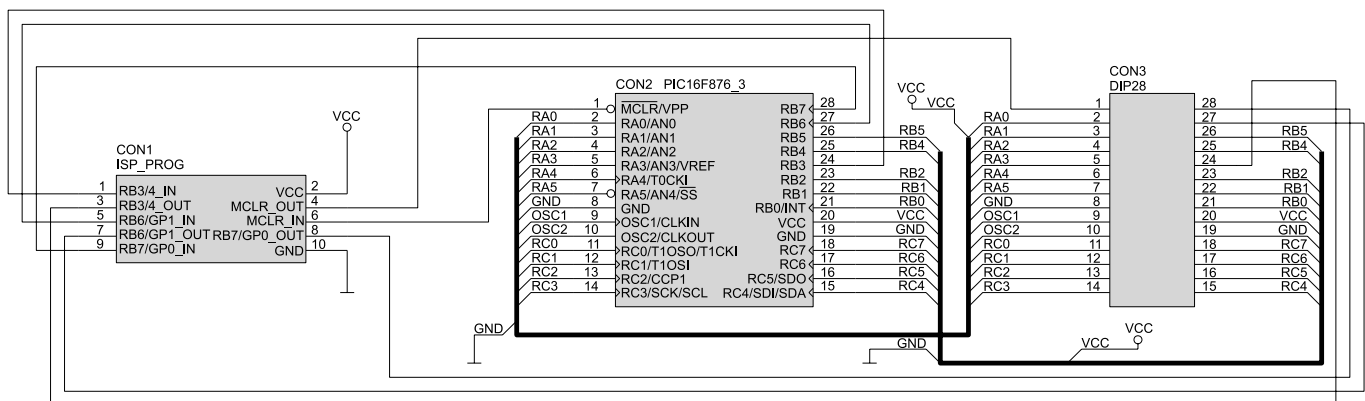
jest zastosowanie dziewięciu podstawek programujących. W tab. 3 przedstawiony jest spis układów przypisanych do konkretnej podstawki programującej. W przypadku podstawek 18-nóżkowych, dla każdej wersji możliwe jest programowanie układów typu PIC16F83, PIC16F84, PIC16F84A. Jest tak, ponieważ podstawki te różnią się między sobą jedynie innym umiejscowieniem nóżki programowania PGM, a wymienione układy nie posiadają takiego wyprowadzenia. Mogą więc być obsługiwane przez dowolną podstawkę. Pozostałe układy przypisane są do konkretnej wersji podstawki.



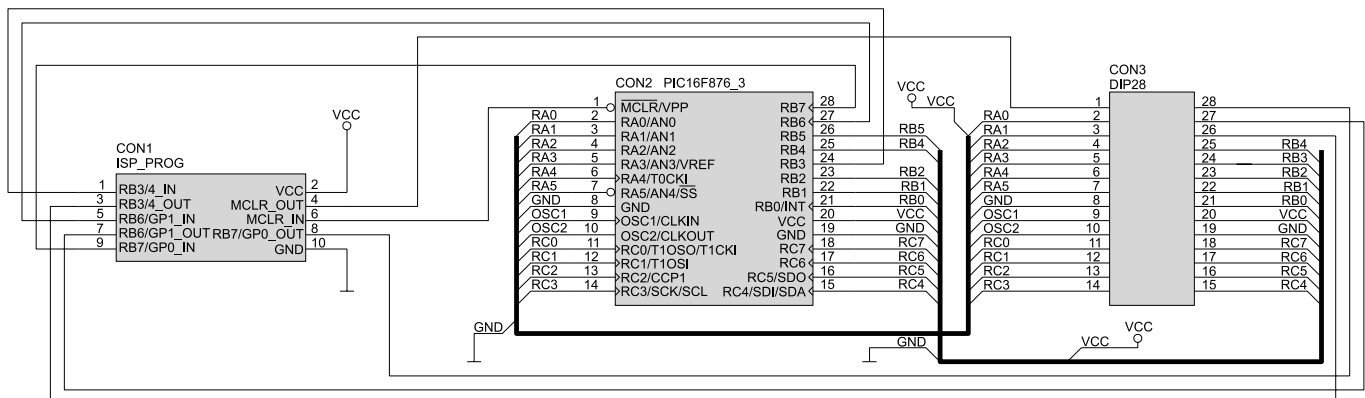
Rys. 18b. Schemat elektryczny adaptera DIP18-2



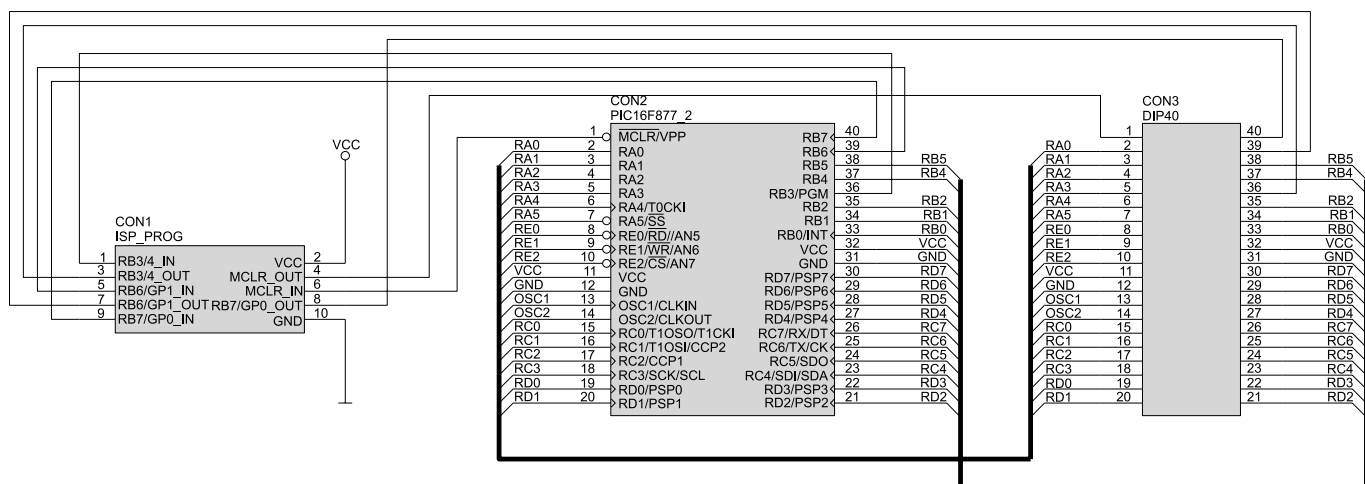
Rys. 18c. Schemat elektryczny adaptera DIP18-3



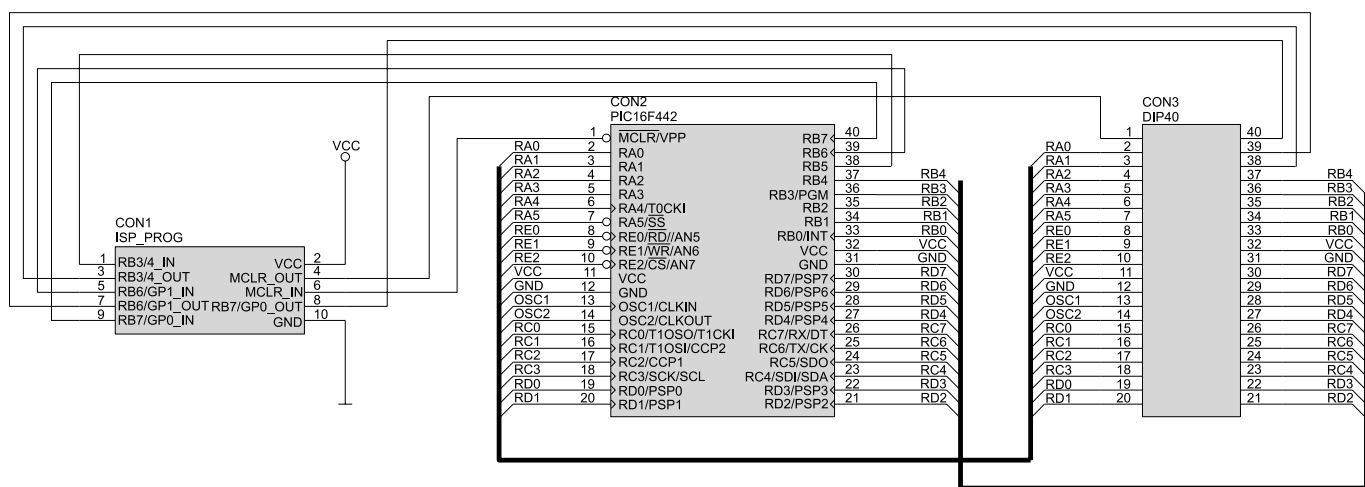
Rys. 19a. Schemat elektryczny adaptera DIP28-1



Rys. 19b. Schemat elektryczny adaptera DIP28-2



Rys. 20a. Schemat elektryczny adaptera DIP40-1



Rys. 20b. Schemat elektryczny adaptera DIP40-2

Charakterystyka podstawek

DIP8 – Schemat elektryczny podstawki DIP8 jest przedstawiony na **rys. 16**. Sygnały ze złącza CON1 kierowane są do podstawki CON2, w której będzie znajdował się mikrokontroler. Sygnały te są sygnałami wejściowymi i na czas programowania będą podłączone do złącza komputera. Ze złącza CON1 sygnały są kierowane również do podstawki CON3, która będzie przyłączona do pracującego systemu. Pozostałe wyprowadzenia podstawek CON2 i CON3, niewykorzystane w procesie programowania, są połączone ze sobą bezpośrednio. W podstawce DIP8 nie jest wykorzystywany sygnał programowania PGM, gdyż te mikrokontrolery nie wymagają takiego sygnału.

DIP14 – Schemat tej podstawki przedstawiony jest na **rys. 17**. Obsługiwane mikrokontrolery

mają bardzo podobną budowę wewnętrzną do układów PIC12F6xx i w zasadzie posiadają tylko większą liczbę wyprowadzeń, dlatego schemat jest analogiczny.

DIP18-1, DIP18-2, DIP18-3 – Te podstawki służą do montowania układów w obudowach 18-nóżkowych. Ze względu jednak na zróżnicowanie wyprowadzeń tych mikrokontrolerów, można wyróżnić trzy grupy układów, dla których wymagana jest odpowiednia podstawa. Schemat elektryczny wszystkich podstawek jest przedstawiony na **rys. 18**. W tym przypadku wykorzystane są wszystkie wyprowadzenia złącza CON1. Różnica pomiędzy grupami mikrokontrolerów polega na innym wyprowadzeniu sygnału PGM, dlatego poszczególne podstawki różnią się innym podłączeniem tego sygnału.

DIP28-1, DIP28-2 – Ten rodzaj podstawek jest podzielony na dwie grupy obsługujące układy z rodziny PIC16 oraz PIC18. Tak jak w przypadku układów 18-nóżkowych, różnica polega na innym umiejscowieniu sygnału PGM. Schemat elektryczny podstawek jest przedstawiony na **rys. 19**.

DIP40-1, DIP40-2 – Podstawki te są podzielone na dwie grupy układów analogicznie jak dla układów 18-nóżkowych: wersja DIP40-1 służy do montażu układów z rodziny PIC16, natomiast wersja DIP40-2 układów z rodziny PIC18. Schemat tych adapterów jest przedstawiony na **rys. 20**.

Krzysztof Pławiuk, EP
krzysztof.plawsiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: pcb.ep.com.pl oraz na płycie CD-EP6/2004B w katalogu PCB.