

Uniwersalny programator mikrokontrolerów PIC, część 1

AVT-573

PROJEKT
Z OKŁADKI

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



Czy budowanie programatora dedykowanego dla układów jednego producenta ma sens? W pierwszej chwili można by sądzić, że raczej niewielki. Zauważmy jednak, że w praktyce często się zdarza, że konstruktorzy przywiązują się właśnie do mikrokontrolerów spod jednego znaku. Dla nich dobry programator może się okazać sporym udogodnieniem w pracy.

Charakterystyka programatora

Opisywany programator umożliwia programowanie mikrokontrolerów PIC zarówno w podstawce programatora, jak i pracującym systemie. Dla programowania w systemie dołączone są specjalne podstawki-przejęciówki umożliwiające normalną pracę mikrokontrolera w stanie spoczynku programatora oraz programowanie bez konieczności jego wyciągnięcia z podstawki. Cały proces programowania odbywa się automatycznie. Istotną cechą programatora jest to, że nie zajmuje on żadnego wyprowadze-

nia mikrokontrolera, wszystkie mogą być wykorzystane do sterowania układami systemu. Moduł programujący został wyposażony w przełączniki elektroniczne w postaci kluczy analogowych, które na czas programowania odłączają potrzebne wyprowadzenia mikrokontrolera od pracującego układu i przyłączają je do programatora, a po zaprogramowaniu wyprowadzenia te są automatycznie przełączane z powrotem do pracującego układu. Dzięki temu nie ma obawy o uszkodzenie systemu w przypadku podania napięcia programującego (12V) na wejście !MCLR mikrokontrolera, do którego może być dołączony układ cyfrowy. W przypadku programowania w podstawce programatora zastosowane klucze analogowe przełączają odpowiednio sygnały sterujące tak, aby niezależnie od typu mikrokontrolera jego pozycja w podstawce była jednakowa (mikrokontroler jest zawsze umieszczony przy lewej krawędzi podstawki). Ułatwia to znacznie proces programowania, szczególnie przy częstym wkładaniu i wyjmowaniu mikrokontrolera nie ma obawy, że zostanie błędnie włożony.

Tab. 1. Podstawowe właściwości programatora

- ✓ Możliwość programowania w systemie
- ✓ Możliwość zasilania z systemu
- ✓ Możliwość zasilania systemu z zasilacza programatora
- ✓ Możliwość pracy przy zasilaniu baterijnym
- ✓ Sygnalizacja świetlna procesu programowania
- ✓ Niski pobór prądu ok. 3 mA w czasie programowania
- ✓ Znikomy pobór prądu w czasie spoczynku <math>< 1 \mu\text{A}</math>
- ✓ Dostępne dwa rodzaje darmowego oprogramowania obsługującego programator
- ✓ Obsługa niemal wszystkich dostępnych mikrokontrolerów PIC
- ✓ Programowanie pamięci z magistralą Microwire

Oprócz programowania mikrokontrolerów programator umożliwia także obsługę pamięci szeregowych typu 93Cxx, wyposażonych w interfejs Microwire. Programowanie pamięci wymaga jednak umieszczenia w odpowiednim miejscu podstawki programującej, gdyż przełączniki nie przełączają sygnałów dla tej pamięci.

Przy projektowaniu programatora szczególnie nacisk został postawiony na maksymalną uniwersalność i wygodę programowania oraz zminimalizowanie pobieranego prądu. W tym celu programator został wyposażony w przełącznik umożliwiający jego zasilanie na trzy sposoby: z zewnętrznego zasilacza, z baterii, korzystając z napięcia dostępnego w programowanym systemie.

Zastosowane układy CMOS sprawiają, że prąd pobierany podczas programowania wynosi około 3mA (jeśli dioda sygnalizacyjna zostanie wyłączona), natomiast prąd spoczynkowy jest mniejszy od 1µA. Tak niewielki pobór prądu umożliwia wielomiesięczną pracę programatora przy zastosowaniu zasilania baterijnego. Dzięki temu przedstawiony programator może być traktowany również jako urządzenie przenośne. Rozbudowany układ zasilania umożliwia także zasilanie uruchamianego systemu z zasilacza programatora, co pozwoli na zmniejszenie potrzebnych źródeł zasilania w czasie prac nad projektem. Najważniejsze funkcje programatora przedstawione są w **tab. 1**.

Oprogramowanie sterujące

Najistotniejszą zaletą przedstawionego programatora jest dostępność darmowego oprogramowania do jego obsługi. Jako aplikację sterującą można wykorzystać jeden z dwóch programów udostępnianych bez żadnych opłat, które można pobrać ze stron autorów. Pierwszy program o nazwie „Oshon PIC Programmer“ został stworzony przez Vladimira Soso i można go pobrać ze strony <http://www.oshonsoft.com/picprog.html>. Oprogramowanie to umożliwia programowanie jedynie mikrokontrolerów z rodziny PIC16. Na stronie tej znajduje się również schemat prostszego programatora, stworzonego dla tego oprogramo-

wania, jednak przedstawiony w artykule programator pomimo znacznego rozbudowania jest zgodny funkcjonalnie z tam przedstawionym.

Drugim programem sterującym jest program IC-Prog stworzony przez Bonny Gijzena, który jest dostępny na stronie <http://www.ic-prog.com>. Ten program posiada dużo większe możliwości i umożliwia programowanie niemalże wszystkich dostępnych obecnie mikrokontrolerów PIC. Może także współpracować z innymi typami programatorów.

Prezentowany układ programatora wraz z odpowiednim oprogramowaniem umożliwia programowanie szerokiej gamy mikrokontrolerów oraz szeregowych pamięci EEPROM. W **tab.2** przedstawiona jest lista programowanych układów. Wymienione zostały jedynie mikrokontrolery z pamięcią programu typu FLASH, pomimo tego, że można programować także z pamięcią EPROM. Z uwagi na fakt, że obecnie stosowane są przede wszystkim mikrokontrolery z pamięcią FLASH, praca programatora została sprawdzona dla tego typu układów.

Budowa

Przedstawiony w niniejszym artykule programator został zbudowany na bazie programatora opracowanego przez Davida Taita o nazwie „Tait 'Classic' Programmer“. Jego schemat jest przedstawiony na **rys. 1**. Programator ten służy głównie do programowania mikrokontrolerów PIC16x8x. Aby przystosować go do programowania pozostałych mikrokontrolerów, schemat ten należało znacznie przekształcić. Porównanie obydwu schematów pozwoli na zobrazowanie wniesionych przeróbek.

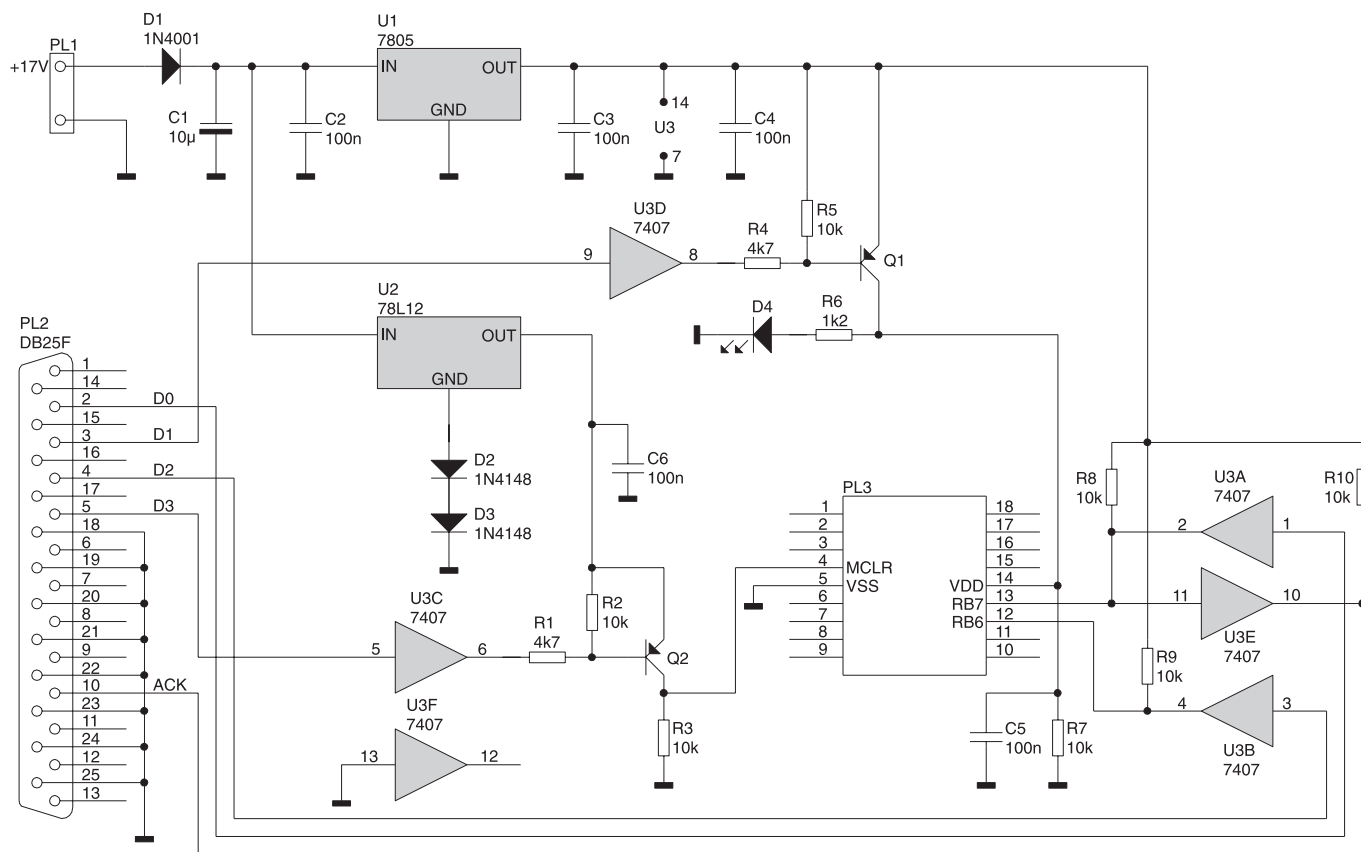
Schemat blokowy zmodyfikowanego programatora jest pokazany na **rys. 2**, zaś schemat elektryczny na **rys. 3**. Cały programator można podzielić na kilka bloków funkcjonalnych: pierwszy blok stanowi interfejs pomiędzy komputerem a programowanym układem. Blok ten składa się z układu US1, który zawiera trzy przełączne klucze analogowe. Interfejs został wykonany w taki sposób, że klucze analogowe zarówno od strony komputera, jak

Tab.2 Spis obsługiwanych mikrokontrolerów z pamięcią programu typu FLASH oraz pamięci EEPROM

L.p.	Typ układu	L.p.	Typ układu
1	PIC12F629	20	PIC16F873
2	PIC12F675	21	PIC16F873A
3	PIC16F72	22	PIC16F874
4	PIC16F73	23	PIC16F874A
5	PIC16F74	24	PIC16F876
6	PIC16F86	25	PIC16F876A
7	PIC16F87	26	PIC16F877
8	PIC16F83	27	PIC16F877A
9	PIC16F84	28	PIC18F242
10	PIC16F84A	29	PIC18F248
11	PIC16F627	30	PIC18F252
12	PIC16F628	31	PIC18F258
13	PIC16F630	32	PIC18F442
14	PIC16F676	33	PIC18F448
15	PIC16F818	34	PIC18F1320
16	PIC16F819	35	PIC18F2320
17	PIC16F870	36	PIC18F4320
18	PIC16F871	37	PIC18F4539
19	PIC16F872	38	93Cxx

i programowanego układu umożliwiają ustawienie na danej linii stanu niskiego. Stan wysoki jest natomiast wymuszany przez rezystory podciągające zawarte w drabince RP1. W ten sposób w stanie logicznej jedynki wyjście odpowiedniego klucza jest w stanie wysokiej impedancji, a stan wysoki jest wymuszany tylko przez zewnętrzne rezystory. Takie rozwiązanie jest szczególnie ważne dla sygnału danych „Data“, gdyż linia ta musi realizować transmisję w dwie strony. Od strony mikrokontrolera linia ta jest podłączona do jednego wyprowadzenia, natomiast od strony komputera sygnał transmisji i odbioru danych jest rozdzielony i dołączony do innych wyprowadzeń złącza LPT. Dzięki temu mikrokontroler może bez problemu wymusić stan niski na tej linii w przypadku, gdy komputer wystawi stan wysoki. Sygnał z komputera jest podawany na wejście sterujące C układu US1, natomiast sygnał powrotny z mikrokontrolera jest podawany na pin 10 złącza równoległego komputera.

Kolejnym blokiem programatora jest przełącznik sygnału zegarowego i sygnału danych. Przełącznik ten jest zbudowany przy użyciu kluczy analogowych zawartych w układzie US2. Prze-



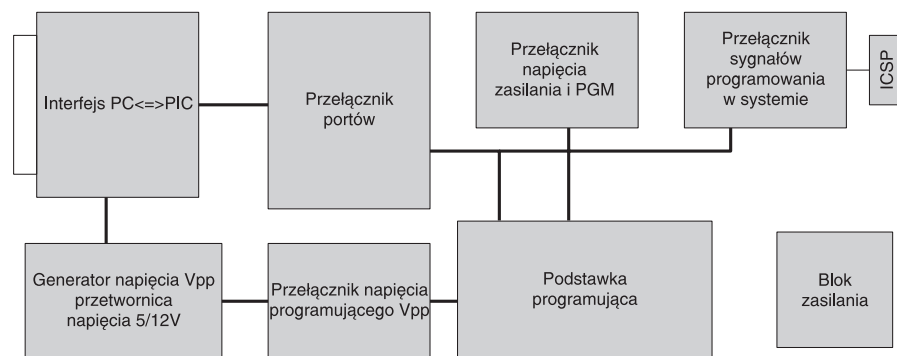
Rys. 1. Schemat programatora Tait 'Classic' Programmer

Łącznik kieruje odpowiednio sygnały do złącza programującego w zależności od programowanego aktualnie mikrokontrolera. Przełączanie to jest konieczne ze względu na różne rozmieszczenie wyprowadzeń programujących w obsługiwanych mikrokontrolerach. I tak dla układów 28- i 40-nóżkowych sygnały te muszą być doprowadzone odpowiednio do wyprowadzenia 39 i 40 złącza CON1. Dla układów w obudowach 18-nóżkowych sygnały te są kierowane do końcówek 34 i 35 bez stosowania przełączników. Problem powstaje w przypadku mikrokontrolerów 8- i 14-nóżkowych, gdyż w tych układach na wyprowadzeniu numer 40 podstawki programatora wymagane jest podanie masy zasilania, a sygnały programujące należy podać na wyprowadzenia 38 i 39. Dlatego potrzebne są jeszcze dodatkowe przełączniki zawarte w kolejnym bloku.

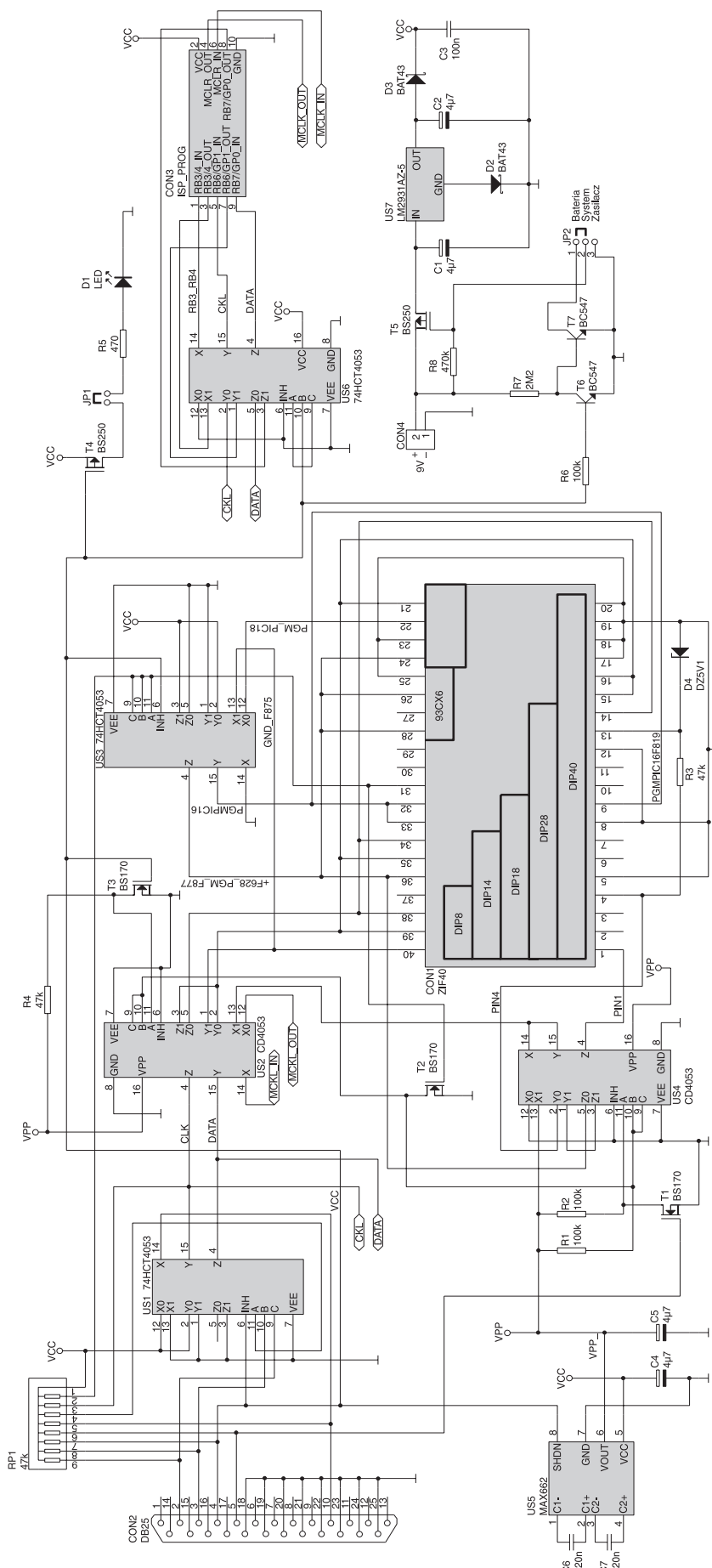
Blok przełączania zasilania i napięcia PGM służy do podania odpowiedniego napięcia zasilającego mikrokontroler oraz na wejście sygnału programowania PGM.

Dla poprawnego procesu programowania wejście PGM powinno być zwarte do masy. Jednak w zależności od mikrokontrolera napięcie to należy podawać na różne wyprowadzenia podstawki programującej. Dla mikrokontrolerów serii PIC18 w obudowach 28- i 40-nóżkowych napięcie to należy podać na port RB5 (nóżka 38 podstawki). W przypadku programowania mikrokontrolerów w obudowach 8- i 14-nóżkowych wyprowadzenie to jest wykorzystywane jako sygnał zegarowy, dlatego za pomocą klucza analogowego „X” zawartego w układzie

US3 napięcie to jest podawane tylko w trybie programowania mikrokontrolerów typu PIC18. Dla mikrokontrolerów typu PIC16 w obudowach 28- i 40-nóżkowych napięcie PGM należy podać natomiast na port RB3 (nóżka 36 podstawki). Również w tym przypadku pojawia się problem, gdyż podczas programowania mikrokontrolera na przykład PIC16F628 nóżka ta stanowi wejście napięcia zasilającego tego układu. Do przełączania napięcia zastosowano klucza „Z” układu US3, który dla układów 28- i 40-nóżkowych podaje masę, a dla układów 18-



Rys. 2. Schemat blokowy programatora uniwersalnego

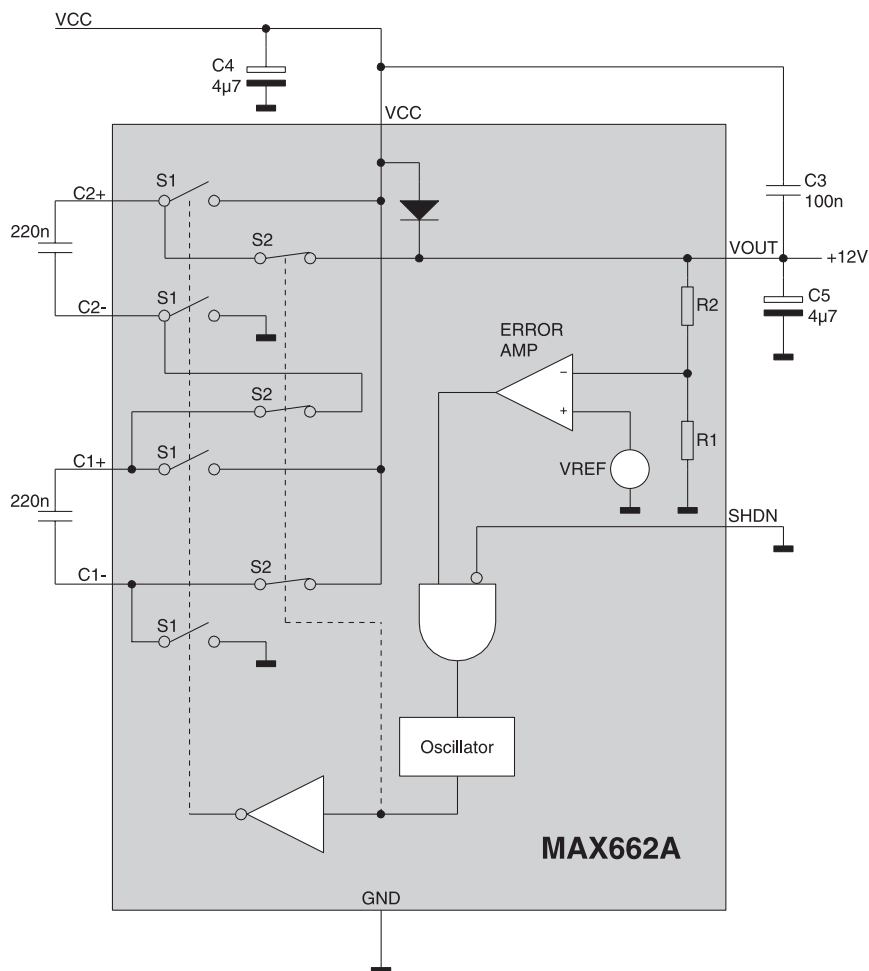


Rys. 3. Schemat programatora mikrokontrolerów PIC (płytką główną)

nóżkowych plus zasilania. Kolejna kolizja powstaje dla nóżki numer 32 podstawki programującej, gdyż dla układów 28- i 40-nóżkowych wyprowadzenie to jest wejściem plusa zasilania, natomiast dla układów 18-nóżkowych PIC16F62x jest wejściem sygnału PGM. Wejście to w czasie programowania musi być na potencjale masy. Do odpowiedniego przełączania tych napięć służy ostatni już klucz układu US3 - „Y”. Dla pozostałych układów odpowiednie sygnały doprowadzone są bezpośrednio do podstawki, bez przełączników.

Również sygnały dla programowania pamięci szeregowej kierowane są bezpośrednio do odpowiednich nóżek podstawki programującej. Wyjątek stanowi sygnał podawany na nóżkę numer 13, gdyż podany tam sygnał pochodzi pośrednio z wyjścia przetwornicy napięcia 5/12 V. Elementy R3 i D4 ograniczają więc jego amplitudę do wartości 5 V.

UWAGA: Rezystancja włączanego klucza analogowego układu US3 wynosi około 100Ω, co powoduje zmniejszenie napięcia zasilającego mikrokontroler. Dla większości mikrokontrolerów spadek ten jest niewielki i nie ma wpływu na proces programowania. Jednak dla układów zawierających dużą pamięć programu (głównie układy PIC18) rezystancja ta obniża napięcie zasilania układu do wartości około 4,6 V. Z uwagi na fakt, że minimalne napięcie zasilania przy programowaniu nie powinno być mniejsze niż 4,5 V, wartość ta jest nieznacznie większa od minimalnej. Może się zdarzyć, że zastosowany egzemplarz układu 74HCT4053 ma nieznacznie większą rezystancję wewnętrzną niż znamionowa i napięcie zasilania mikrokontrolera będzie mniejsze od wymaganego. Aby temu zapobiec, jako układ US3 należy zastosować dwa układy 74HCT4053 połączone równolegle (jeden przylutowany do drugiego i włożony w podstawkę). Spowoduje to zmniejszenie rezystancji wewnętrznej kluczy i wzrost napięcia zasilającego mikrokontroler, dla wymienionego przypadku napięcie to będzie wynosiło 4,8 V. Montaż drugiego układu dotyczy skrajnego przypadku, gdzie układ



Rys. 4. Budowa wewnętrzna przetwornicy napięcia MAX662

US3 ma rezystancję wewnętrzną większą niż standardowo założoną przez producenta. W modelowym układzie sprawdzona została praca dla kilku egzemplarzy układu 74HCT4053 i wystarczające było zastosowanie pojedynczego układu.

Kolejnym blokiem jest przetwornica napięcia oraz układ przełączników napięcia programującego (V_{pp}). Jako generator niezbędnego napięcia o wartości 12 V zastosowano specjalizowaną przetwornicę typu MAX662. Budowa wewnętrzna przetwornicy jest przedstawiona na rys. 4. Układ ten jest wysoko sprawny przetwornicą umożliwiającą uzyskanie napięcia 12 V przy zasilaniu napięciem 5 V. Maksymalna wydajność prądowa wynosi 30 mA, przy tym prąd pobierany przez przetwornicę mieści się w granicach kilkuset mikroamperów. Dodatkowo przetwornica posiada wejście SHDN umożliwiające przełączenie jej w tryb czuwania, ogra-

niczając pobierany prąd do wartości około 0,5 μ A. Ta właściwość jest szczególnie ważna przy baterijnym zasilaniu programatora. Napięcie V_{pp} wytworzone w przetwornicy kierowane jest następnie do przełącznika zbudowanego z układu US4, który w zależności od programowanego układu kieruje to napięcie na nóżkę 1 lub na nóżkę 4 podstawki programującej. Dla mikrokontrolerów umieszczonych w obudowach 28- i 40-nóżkowych napięcie to należy podać na nóżkę 1, natomiast umieszczonych w obudowach 8-, 14- i 18-nóżkowych na nóżkę 4. W przypadku mikrokontrolerów 8- i 14-nóżkowych dodatkowo na wyprowadzenie 1 należy podać plus zasilania. Funkcje przełączania tych sygnałów pełni układ US4.

Do automatycznego przełączania obwodów została wykorzystana właściwość mikrokontrolerów umieszczonych w obudowach 28- i 40-nóżkowych. W układach tych

masę należy podać na dwa wyprowadzenia podstawki - 12 i 31. Jednak obydwa wyprowadzenia są zwarte wewnętrznie i masa została doprowadzona tylko do wejścia numer 12. W tej sytuacji włożenie w podstawkę układu w obudowie 28- lub 40-nóżkowej spowoduje pojawienie się stanu niskiego na wyprowadzeniu 31. Na tej podstawie „wykrywany” jest rodzaj włożonego mikrokontrolera. Dla mikrokontrolerów 28- i 40-nóżkowych na wyprowadzeniu 31 panuje stan niski, dla pozostałych natomiast stan wysoki wymuszony przez rezystor zawarty w drabince RP1. W zależności od tego stanu klucze analogowe zawarte w układach US2, US3, US4 przełączają odpowiednio sygnały sterujące. Takie rozwiązanie jest znacznie bezpieczniejsze niż programowy wybór rodzaju mikrokontrolera (program IC-Prog oferuje taką możliwość). Chodzi tutaj głównie o napięcie programujące (V_{pp}), gdyż przełączanie pomiędzy dwoma wyprowadzeniami układu mogłoby go uszkodzić w przypadku wybrania w programie innego układu niż włożony w podstawkę programującej. W przypadku zastosowanego przełącznika takie ryzyko nie występuje, gdyż przy braku mikrokontrolera w podstawce lub posiadającego mniejszą liczbę wyprowadzeń niż 28, napięcie programujące zawsze będzie przyłożone do wyprowadzenia numer 4. Jeśli zostanie włożony większy układ, to napięcie to zawsze będzie przyłożone do wyprowadzenia numer 1. Dodatkowo w stanie spoczynku wszystkie przełączniki (klucze analogowe) są w stanie wysokiej impedancji, dlatego podczas wkładania i wyjmowania mikrokontrolera od podstawki odłączone jest napięcie zasilania i programujące. Przełączniki oraz przetwornica napięcia uaktywniane są tylko na czas programowania, co z kolei pozwala na ograniczenie prądu pobieranego w stanie spoczynku przez programator.

Ostatnim blokiem jest przełącznik sygnałów do programowania w systemie. Blok ten składa się z układu US6 oraz jednego klucza zawartego w układzie US2. Blok ma na celu przełączanie wyprowadzeń mikrokontrolera pracującego w systemie. W czasie

normalnej pracy poprzez klucze wszystkie wyprowadzenia mikrokontrolera przyłączone są do systemu, w którym pracuje. Na czas programowania niezbędne wyprowadzenia (RB7, RB6, PGM, MCLR/Vpp) przyłączane są do programatora, a po zaprogramowaniu automatycznie powracają do układu. Po zaprogramowaniu mikrokontroler zostaje automatycznie wyzerowany i rozpoczyna pracę według nowo zapisanego programu. Taki sposób programowania umożliwia dowolne wykorzystanie wyprowadzeń mikrokontrolera w systemie. Jeżeli wyprowadzenia nie byłyby przełączane, to mogłoby się tak zdarzyć, że na przykład wyprowadzenie RB7 jest obciążone rezystorem o niewielkiej wartości (kilkaset ohmów) i programator nie mógłby wymusić odpowiednich stanów na tym wejściu. Bardziej niebezpieczną sytuacją jest podanie napięcia programującego na wejście !MCLR, które w większości mikrokontrolerów może być wykorzystane jako wejście cyfrowe. Przyłożenie napięcia o wartości 12V mogłoby uszkodzić współpracujące układy. Zastosowanie przełącznika portów całkowicie zabezpiecza przed takimi sytuacjami.

Zastosowanie jako przełączników kluczy analogowych ma jednak też swoje wady, które w sporadycznych przypadkach mogą ograniczać możliwości mikrokontrolera. Wadą wszystkich kluczy analogowych jest fakt, że w stanie włączenia ich rezystancja nie jest równa zero. Zastosowane klucze (US6) charakteryzują się wewnętrzną rezystancją w stanie przewodzenia na poziomie 100 Ω . Jeśli wyprowadzenie mikrokontrolera steruje na przykład diodą LED, to dodatkowa rezystancja ograniczy prąd płynący w obwodzie. Jednak w większości przypadków taka wartość rezystancji nie będzie miała wpływu na funkcjonowanie systemu.

W programatorze zostały zastosowane dwa rodzaje kluczy analogowych, różniące się napięciem zasilania oraz rezystancją wewnętrzną. Dla przełączania sygnałów o wartościach 0...5 V zastosowano układy typu 74HCT4053 posiadające rezystancję wewnętrzną o wartości około 100 Ω . Układy

te nie nadają się jednak do przełączania napięcia programującego, gdyż maksymalne napięcie ich zasilania wynosi 5 V. Do przełączania napięcia programującego (12 V) zastosowane zostały układy serii CD4053 umożliwiające prace przy napięciu zasilającym 12 V, niestety posiadają większą rezystancję wewnętrzną, na poziomie 400 Ω . Jednak w przypadku napięcia Vpp rezystancja ta nie ma większego znaczenia, gdyż prąd płynący w tym obwodzie jest niewielki.

Jako wskaźnik programowania została zastosowana dioda świecąca D1, która może zostać wyłączona lub włączona poprzez zworę JP1. Umożliwia to ograniczenie pobieranego prądu przez programator w przypadku zasilania baterijnego lub z pracującego systemu.

Do zasilania programatora został wykonany układ stabilizatora z automatycznym włącznikiem. Dzięki temu w czasie spoczynku programator może być całkowicie odłączony od źródła zasilania i nie pobiera wtedy prądu. Napięcie zasilania należy podłączyć do złącza CON4. Napięcie to trafia dalej na źródło tranzystora T5. W zależności od ustawienia zworki JP2, tranzystor T5 może być sterowany na różne sposoby:

1. Zasilanie z baterii. Jeżeli zworka JP2 będzie ustawiona w pozycji 2-3, to będzie to tryb zasilania baterijnego. Wtedy w czasie spoczynku programator jest całkowicie odłączony od baterii. W chwili rozpoczęcia procesu programowania na wyjściu 4 złącza CON2 pojawi się stan niski i w konsekwencji poprzez tranzystor T7 zostanie wysterowana bramka tranzystora T5, a programator zostanie zasilony. Po zaprogramowaniu mikrokontrolera wyjście 4 złącza CON2 zmieni stan na wysoki i napięcie zasilania zostanie odłączone. Tryb ten może być również ustawiony przy zasilaniu z zasilacza.

2. Zasilanie z systemu - zworka JP2 jest całkowicie rozwarta. Tryb ten można wykorzystać, jeśli programator będzie przyłączony do pracującego systemu poprzez złącze CON3. W takiej sytuacji zewnętrzny zasilacz jest zbędny,

gdyż programator wykorzystuje napięcie zasilające mikrokontroler.

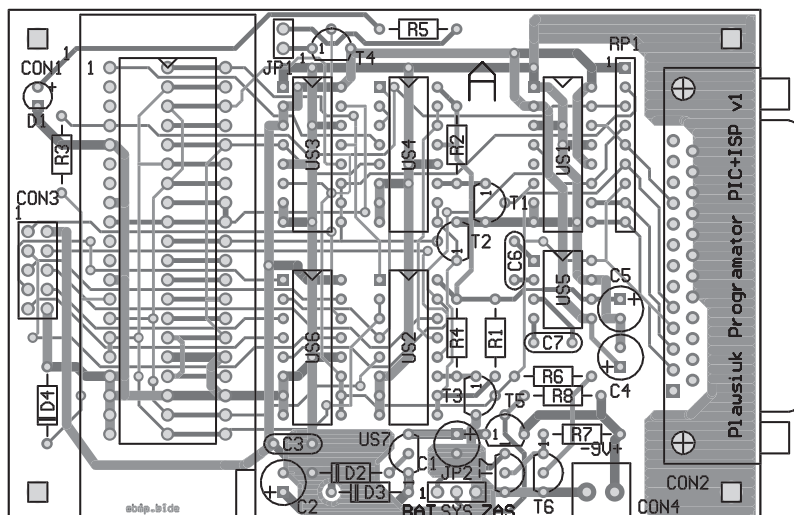
3. Zasilanie z zasilacza - zworka w pozycji 1-2. W tej sytuacji wszystkie układy programatora są zasilane przez cały czas. Jeżeli programator będzie programował mikrokontroler umieszczony w systemie, to możliwe jest zasilanie tego systemu z zasilacza programatora. Maksymalny prąd pobierany nie może być większy niż 50 mA (większy pobór może uszkodzić stabilizator napięcia US7 lub tranzystor T5), jednak dla większości prostych układów są to wartości wystarczające. Pozwoli to na zmniejszenie liczby doprowadzanych przewodów zasilających.

Uwaga: Jeżeli programator służy do programowania mikrokontrolera w systemie, który posiada własny zasilacz, to należy odłączyć zasilanie zewnętrzne od programatora.

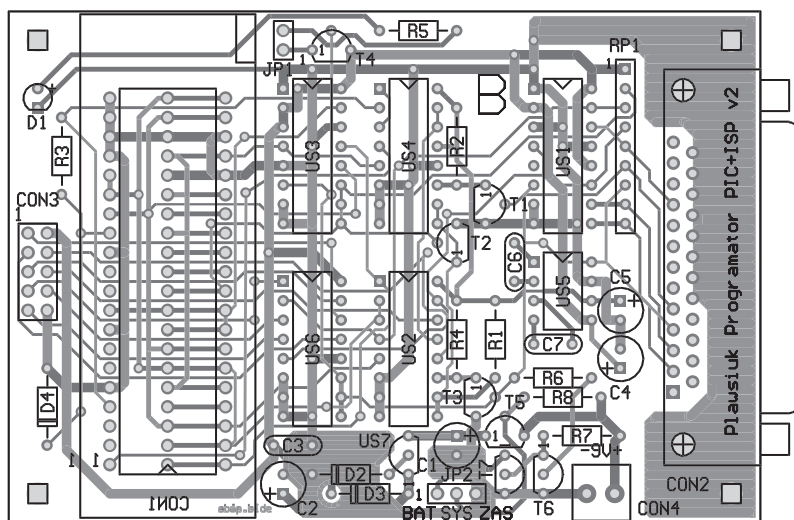
Aby zabezpieczyć wyjście stabilizatora przed uszkodzeniem napięciem z pracującego systemu (w przypadku programowania w systemie), została zastosowana dioda D3, która uniemożliwia przepływ prądu w stronę stabilizatora. W przypadku zasilania z zasilacza dioda ta powoduje spadek napięcia. Aby zrekompensować ten spadek, został przesunięty potencjał masy stabilizatora poprzez diodę D2. W efekcie, na wyjściu stabilizatora utrzymuje się napięcie o wartości 5,4V, natomiast na katodzie diody D3 wymagane napięcie 5V.

Montaż

Programator może zostać zmontowany na jednej z dwóch dostępnych płytek. Jedna płytka umożliwia umieszczenie podstawki programującej po stronie elementów - wersja A (**rys. 5**), natomiast druga wersja płytki umożliwia montaż podstawki programującej po stronie lutowania - wersja B (**rys. 6**). Pierwszy rodzaj montażu jest wskazany, jeśli programator nie będzie umieszczony w obudowie, wtedy płytkę można umieścić na nóżkach, a programowane mikrokontrolery wkładać od góry (od strony elementów). Jeśli programator ma być umieszczony w obudowie, to wygodniejszym rozwiązaniem jest zastosowanie



Rys. 5. Rozmieszczenie elementów na płytce programatora przystosowanej do montażu podstawki programującej od strony elementów (wersja A)



Rys. 6. Rozmieszczenie elementów na płytce programatora przystosowanej do montażu podstawki programującej od strony lutowania (wersja B)

płytki w wersji B (rys. 6), gdyż płytkę można przykręcić do obudowy od strony lutowania i wykonać otwór na podstawkę programującą. Montaż elementów w obu przypadkach wygląda tak samo, oprócz montażu podstawki programującej. Czynności należy rozpocząć od wlutowania rezystorów, następnie podstawek pod układy scalone (bez podstawki programującej). W dalszej kolejności należy wlutować tranzystory, diodę świecącą, zworki JP1 i JP2 oraz złącza. Podstawkę programującą w zależności od wersji płytki montuje się od strony elementów lub od strony lutowania. Obydwie płytki przystosowane są do montażu uniwersalnej podstawki typu ZIF40 (uniwersal-

na 300 i 600 mils) lub zwykłej podstawki precyzyjnej. Ponieważ podstawka musi umożliwiać montaż układów szerokości zarówno 600 mils (DIP40), jak również wąskich 300 mils (DIP8, DIP14, DIP18, DIP28) w płytce zostały wykonane dwa rzędy otworów: o szerokości 300 mils oraz 600 mils. W przypadku zastosowania podstawki typu ZIF montuje się ją w otwory o szerszym rozstawie, a podstawka umożliwia włożenie obydwu rodzajów układów. W przypadku zastosowania podstawki precyzyjnej należy zastosować trzy listwy precyzyjne i wlutować je w każdy rząd otworów. Układy 40-nóżkowe będą wkładane w listwy rozstawione szerzej, natomiast układy w mniejszych

WYKAZ ELEMENTÓW

Płytką główną

Rezystory

R1, R2, R6: 100k Ω

R3...R5: 47k Ω

R7: 2,2M Ω

R8: 470k Ω

Kondensatory

C1, C2: 4,7 μ F/16V

C3: 100nF

C4, C5: 4,7 μ F/16V

C6, C7: 220nF

Półprzewodniki

D1: LED 3mm czerwona

D2, D3: BAT43

D4: Dioda Zenera 5,1V

T1...T3: BS170

T4, T5: BS250

T6, T7: BC547

US1, US3, US6: 74HCT4053

US2, US4: CD4053 (HEF4053)

US5: MAX662

US7: LM2931AZ-5

Różne

CON1: nie wchodzi w skład zestawu i należy zamówić oddzielnie, opis w tekście

CON2: DB25 męskie kątowe

CON3: goldpin 5x2 męskie

obudowach należy wkładać w listwy o węższym rozstawie. Po zmontowaniu wszystkich elementów należy włożyć układy scalone w podstawki. W podstawkę US3 należy włożyć jeden układ 74HCT4053, a w zależności od potrzeb po uruchomieniu całego programatora dolutować do niego równolegle drugi układ, według wcześniejszego opisu (drugi egzemplarz układu 74HCT4053 nie wchodzi w skład kitu). Złącze CON2 należy połączyć kablem ze złączem drukarkowym (LPT) komputera, a do złącza CON4 dołączyć napięcie zasilania o wartości około 9 V. Zworę JP2 należy ustawić w pozycji „Zasilacz“, po czym można przejść do konfiguracji oprogramowania.

Krzysztof Plawsiuk, 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-EP5/2004B w katalogu PCB.

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.

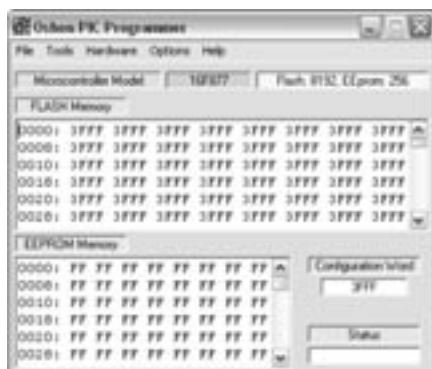
Oprogramowanie

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-



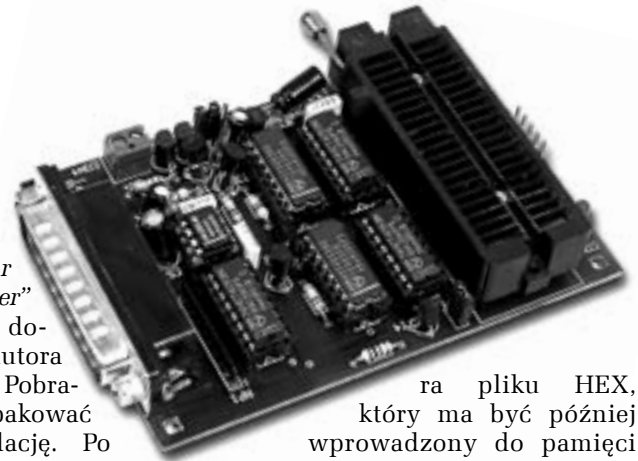
Rys. 7. Wygląd okna Oshon PIC programmer



Rys. 8. Okno konfiguracji linii portu LPT



Rys. 9. Okno testowania programatora

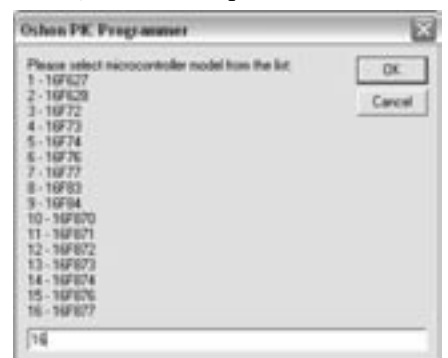


ra pliku HEX, który ma być później wprowadzony do pamięci mikrokontrolera.

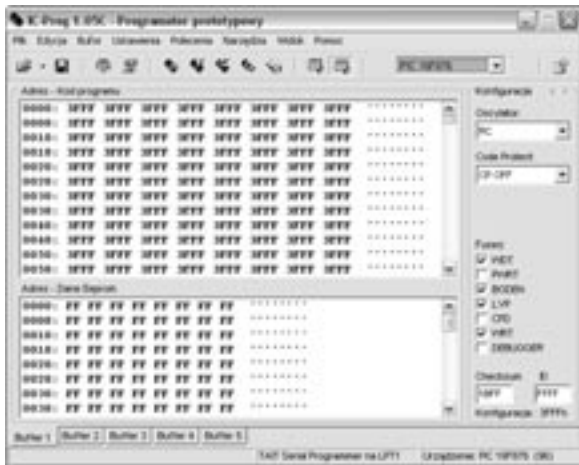
Tools – w tym menu zawarte są polecenia zapisu, odczytu i weryfikacji pamięci.

Hardware – menu to dotyczy parametrów dołączonego programatora. Opcja *Schematics* wyświetla okno ze schematem prostego programatora współpracującego z oprogramowaniem. W zależności od zastosowanego układu pełniące 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

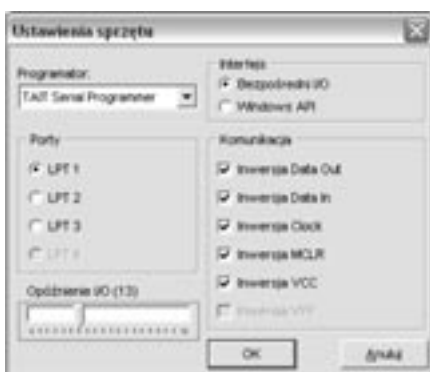


Rys. 11. Wygląd okna programu ICProg

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.

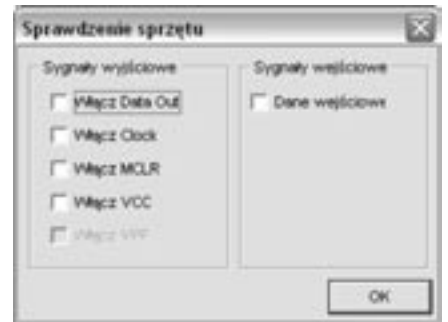
Programator IC-Prog

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-

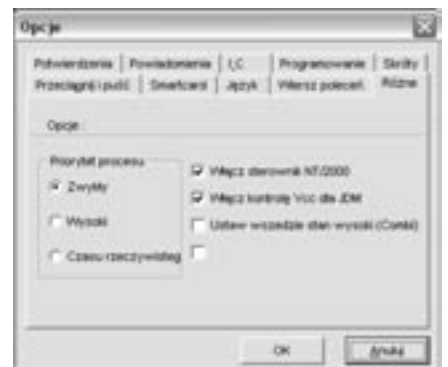


Rys. 13. Okno testowania dołączonego programatora

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-*

lecenie 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.

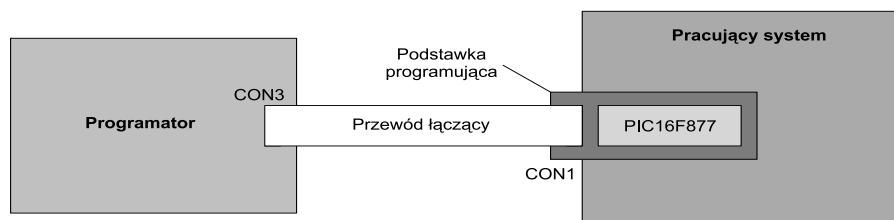
Przykładowa komenda programowania może być następująca:

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

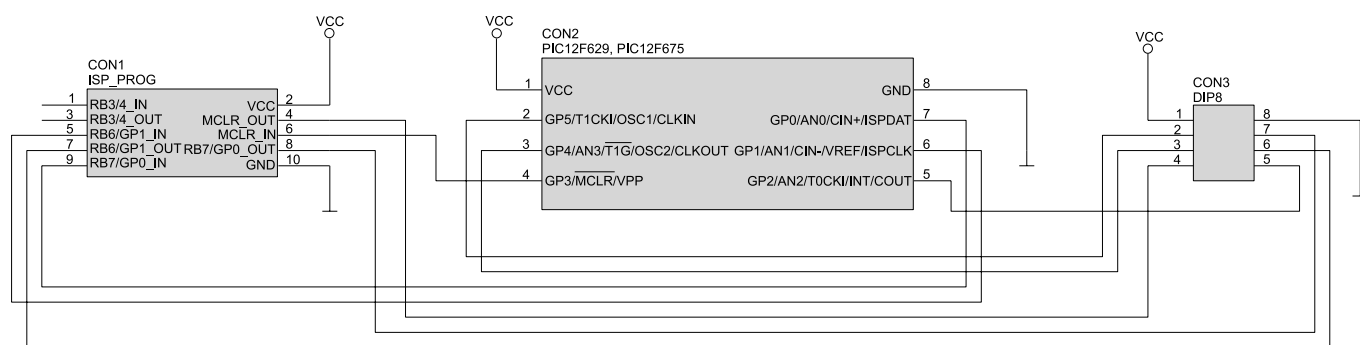
Po wydaniu takiej komendy zostanie uruchomiony IC-Prog, następnie będzie wczytany plik test.hex, którym zostanie zaprogramowany mikrokontroler, dalej zostanie zweryfikowana poprawność zapisu i zapisane słowo konfiguracyjne. Okno programatora będzie wyświetlone w postaci zminimalizowanej (tylko wskaźnik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

Tab. 3. Zestawienie mikrokontrolerów obsługiwanych przez odpowiednie podstawki programujące

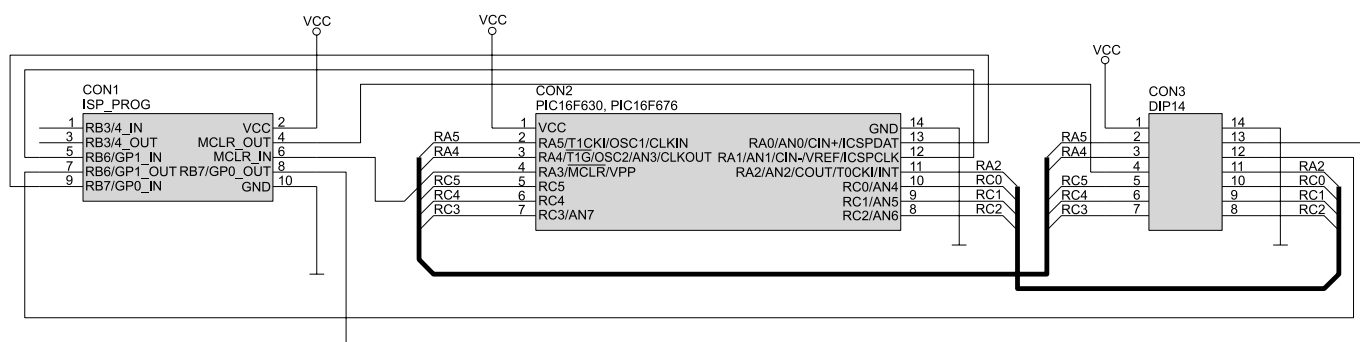
Podstawka	Obsługiwane układy
DIP8	PIC12F629, PIC12F675
DIP14	PIC16F630, PIC16F676
DIP18-1	PIC16F83, PIC16F84, PIC16F84A, PIC16F627, PIC16F628
DIP18-2	PIC16F83, PIC16F84, PIC16F84A, PIC16F818, PIC16F819
DIP18-3	PIC16F83, PIC16F84, PIC16F84A, PIC18F1320
DIP28-1	PIC16F72, PIC16F73, PIC16F76, PIC16F870, 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



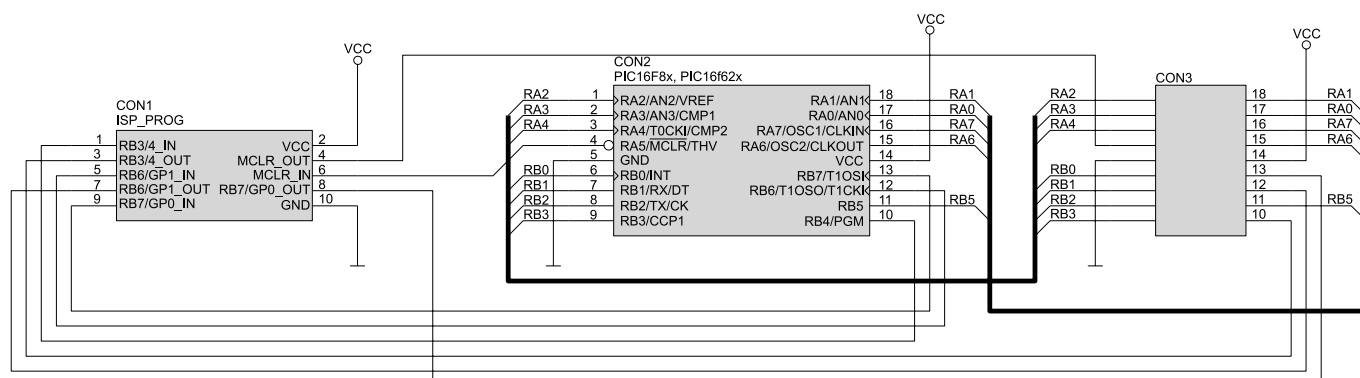
Rys. 15. Sposób podłączenia programatora do pracującego systemu



Rys. 16. Schemat elektryczny adaptera DIP8



Rys. 17. Schemat elektryczny adaptera DIP14



Rys. 18a. Schemat elektryczny adaptera DIP18-1

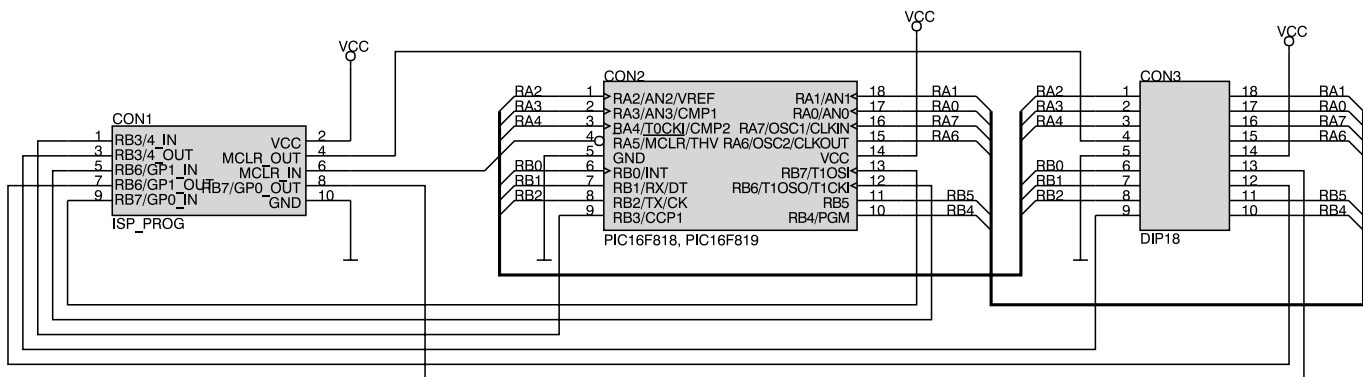
dać, taki tryb programowania jest bardzo przydatny w przypadku prac nad projektem, natomiast przy programowaniu plikami wynikowymi wygodniejszy będzie standardowy sposób programowania za pomocą klawiszy funkcyjnych lub myszki.

Akcesoria dodatkowe

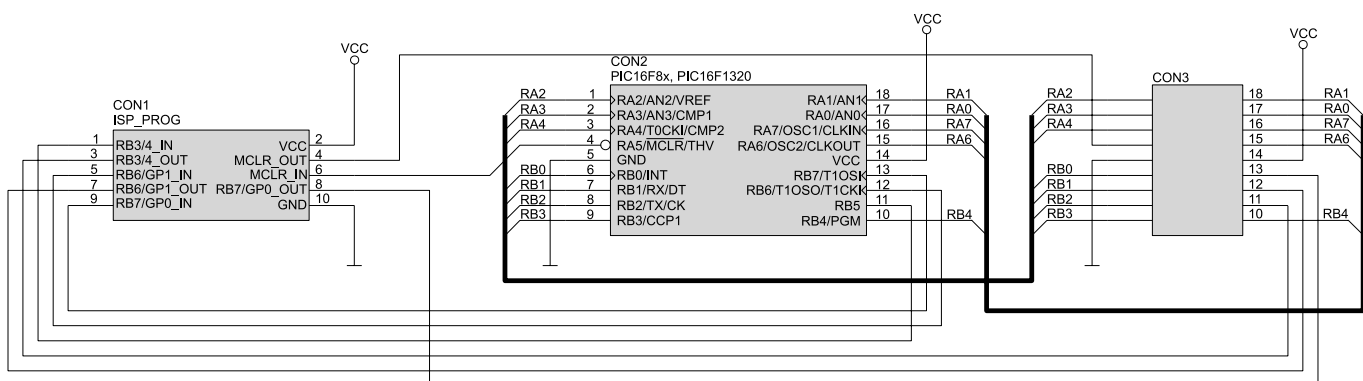
Do programowania w systemie należy zastosować dodatkowe podstawki programujące, które przyłączane są do programatora poprzez złącze CON3. W zależności od typu mikrokontrolera należy zastosować odpowiednią podstawkę. Ze względu na podobieństwa pew-

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

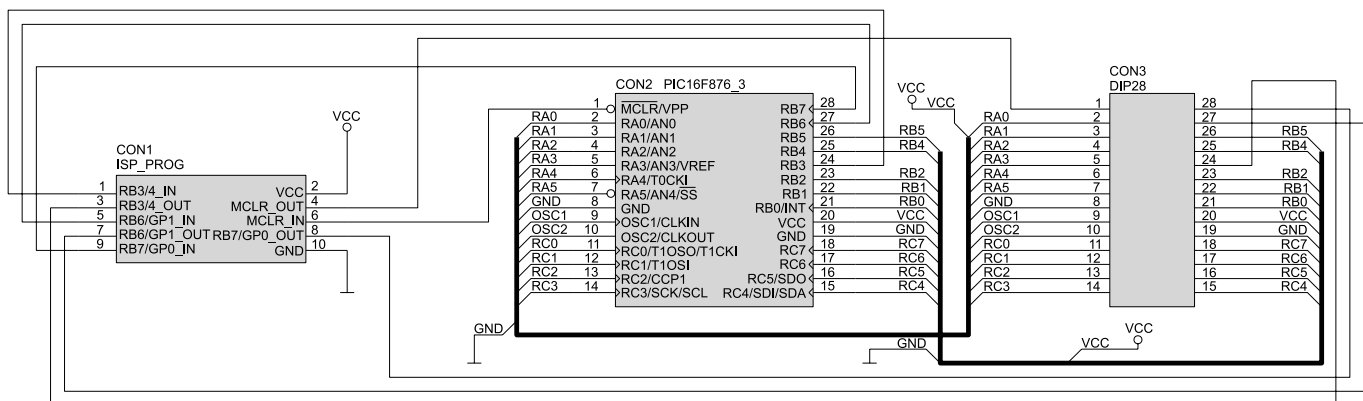
jest zastosowanie dziewięciu podstawek programujących. W tab. 3 przedstawiony jest spis układów przypisanych do konkretnej podstawki programującej. W przypadku podstawek 18-nóżkowych, dla każdej wersji możliwe jest programowanie układów typu PIC16F83, PIC16F84, PIC16F84A. Jest tak, ponieważ podstawki 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.



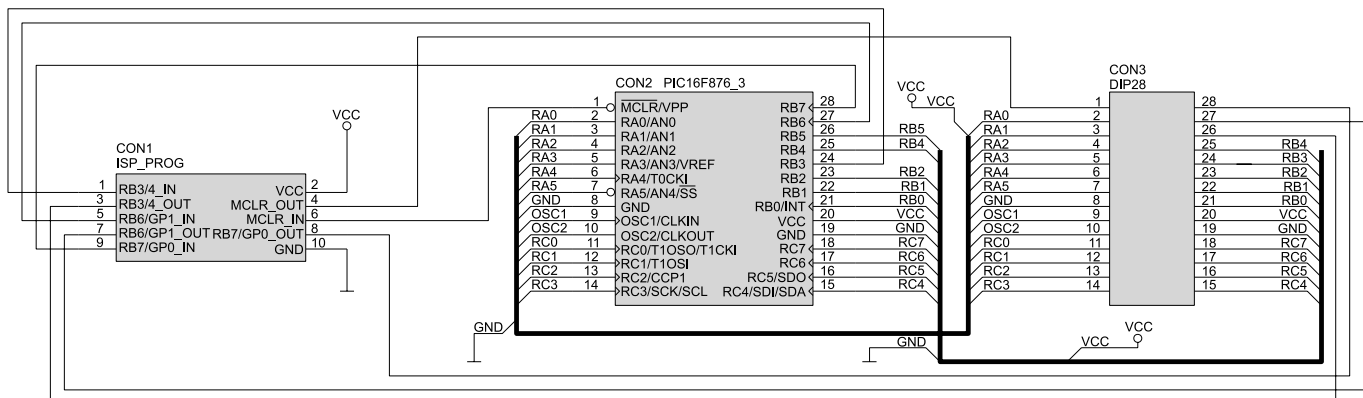
Rys. 18b. Schemat elektryczny adaptera DIP18-2



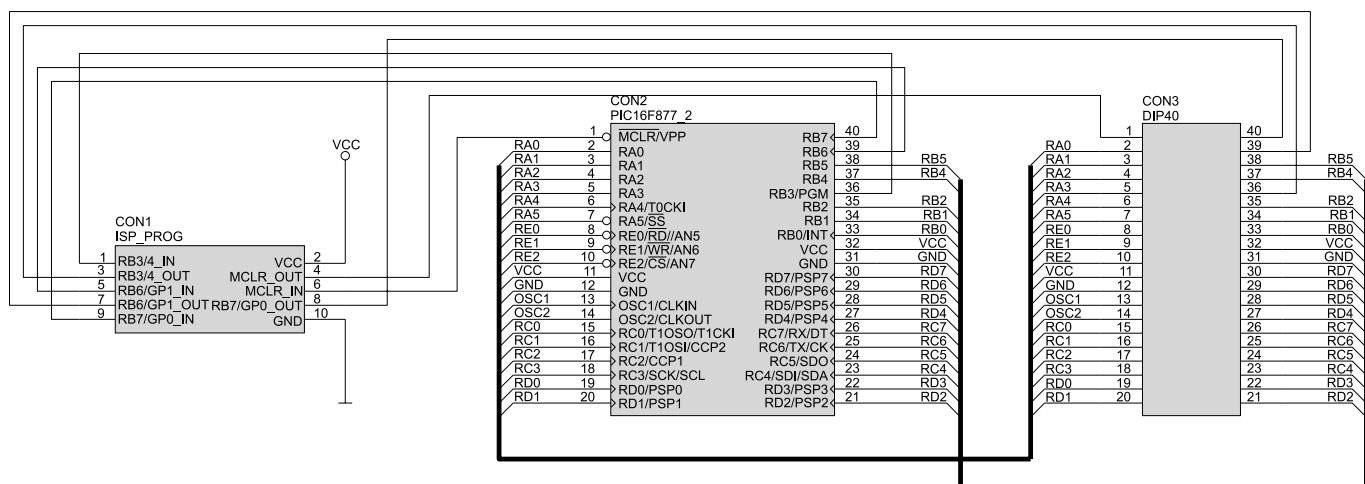
Rys. 18c. Schemat elektryczny adaptera DIP18-3



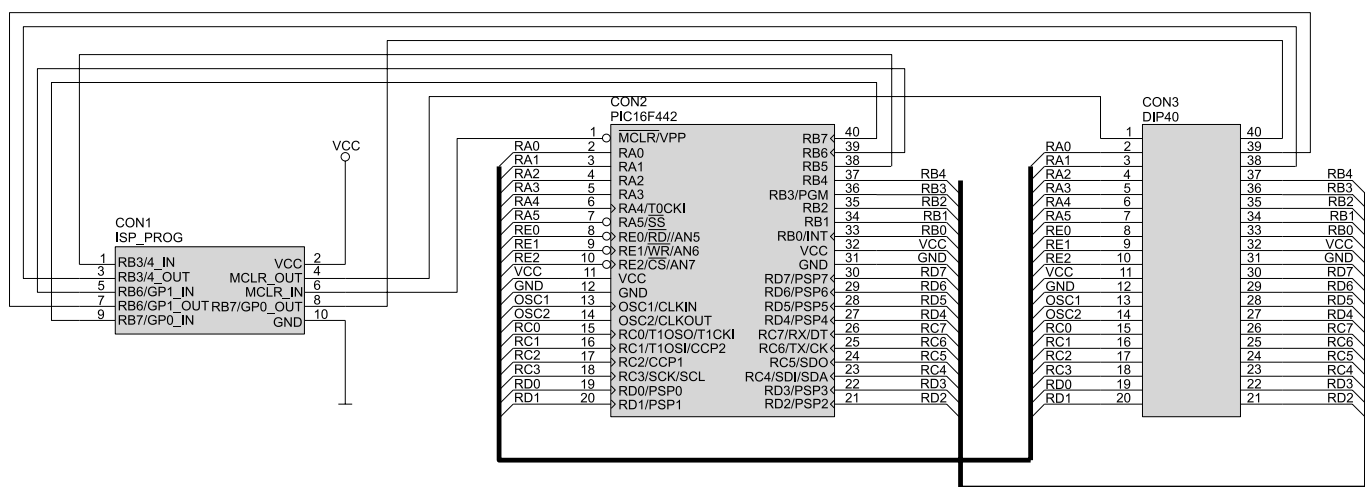
Rys. 19a. Schemat elektryczny adaptera DIP28-1



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.

Uniwersalny programator mikrokontrolerów PIC, część 3

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.

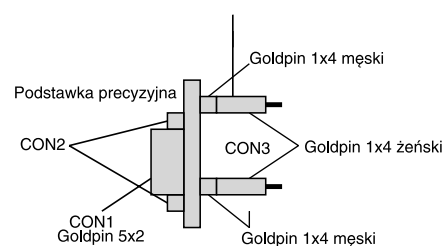
Rekomendacje: Przedstawiony w artykule układ jest uniwersalnym systemem umożliwiającym programowanie mikrokontrolerów firmy Microchip.

Montaż

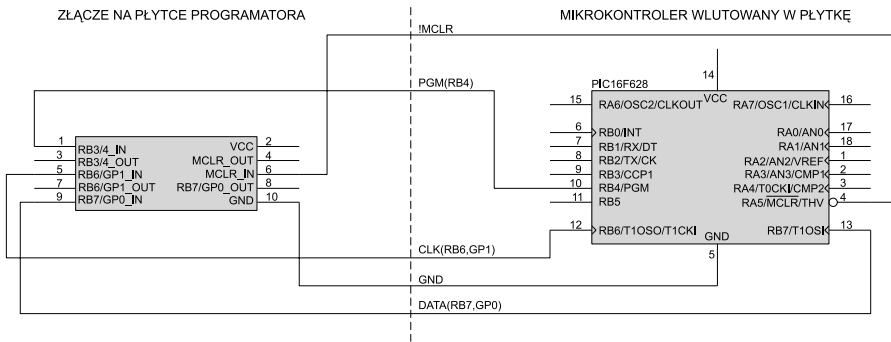
Montaż wszystkich podstawek przebiega jednakowo. Podstawki CON2 są zwykłymi podstawkami pod układ scalony, natomiast podstawa CON3 składa się z dwóch elementów: złącza męskiego typu Goldpin wlotowanego w płytke oraz takiego samego złącza żeńskiego. Obydwa złącza są połączone ze sobą zgodnie z rys. 21. Złącze żeńskie jest typu precyzyjnego i końcówki, które w normalnej sytuacji służą do wlotowania w płytke, w tym przypadku są wkładane w podstawkę systemu. Końcówki te mają średnicę 0,5 mm i dlatego bez problemu można je włożyć zarówno w zwykłe, jak i precyzyjne podstawki pod układy scalone. Zastosowanie tylko złącza męskiego uniemożliwiłoby montaż w podstawie precyzyjnej, gdyż złącze to ma średnicę około 0,7 mm, a w przypadku standardowej podstawki spowodowałoby zbytne rozgięcie jej styków i uniemożliwienie wmontowania mikrokontrolera, który wypadłby z niej.

Przewód połączeniowy programatora z podstawkami należy wykonać z przewodu taśmowego 10-żyłowego o długości około 20 cm. Należy zakończyć go złączami

typu FC-10. Przy montażu tych złączy należy zwrócić uwagę, aby po połączeniu, końcówka numer 1 w złączu CON3 programatora była połączona z końcówką złącza CON1 podstawki programującej. Przygotowanie do programowania w systemie polega na włożeniu mikrokontrolera w podstawkę CON2, następnie w podstawkę systemu złącze CON3 i połączeniu tej podstawki wykonanym kablem ze złączem CON3 programatora. Następnie w programatorze należy wybrać sposób jego zasilania (według wcześniejszego opisu) i po tej czynności można przejść do pracy nad tworzonym projektem bez konieczności wyjmowania mikrokontrolera przy każdorazowym programowaniu.



Rys. 21. Sposób montażu złączy w podstawkach programujących

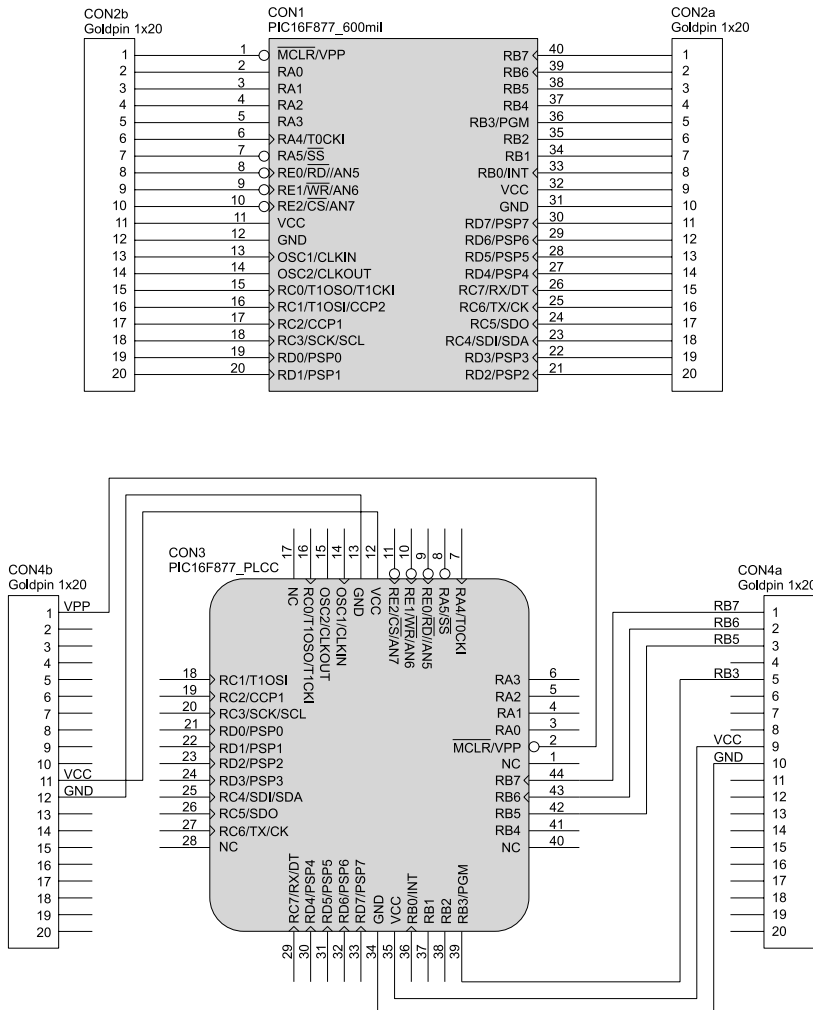


Rys. 22. Schemat połączenia programatora z mikrokontrolerem wlotowym w płytce

Kabel do programowania układów SMD w systemie

Zastosowanie mikrokontrolera w obudowie SMD uniemożliwia programowanie w sposób analogiczny jak to ma miejsce w przypadku układów umieszczonych w obudowach DIP. Wlotowanie układu w płytke nie pozwala na przełączanie jego wyprowadzeń pomiędzy systemem i programatorem, dlatego możliwe jest tylko równoległe przyłączenie sygnałów z programatora. Do progra-

mowania zostaną wykorzystane sygnały ze złącza CON3 programatora, tak jak w przypadku zastosowania podstawek programujących, jednak w ograniczonym stopniu. Przy normalnej pracy mikrokontrolera programator jest odłączony, a tylko na czas programowania klucze analogowe zawarte w układzie US6 przyłączają do mikrokontrolera odpowiednie sygnały. Sposób przyłączenia mikrokontrolera do programatora jest przedstawiony na rys. 22.



Rys. 23. Schemat elektryczny przejściówki DIP40<->PLCC44

WYKAZ ELEMENTÓW

Adapter DIP8

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP8
- CON3: goldpin 8x1 męski + goldpin 8x1 żeński, precyzyjny

Adapter DIP14

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP14
- CON3: goldpin 14x1 męski + goldpin 14x1 żeński, precyzyjny

Adapter DIP18-1

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP18
- CON3: goldpin 18x1 męski + goldpin 18x1 żeński, precyzyjny

Adapter DIP18-2

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP18
- CON3: goldpin 18x1 męski + goldpin 18x1 żeński, precyzyjny

Adapter DIP18-3

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP18
- CON3: goldpin 18x1 męski + goldpin 18x1 żeński, precyzyjny

Adapter DIP28-1

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP28 300mils
- CON3: goldpin 28x1 męski + goldpin 28x1 żeński, precyzyjny

Adapter DIP28-2

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP28 300mils
- CON3: goldpin 28x1 męski + goldpin 28x1 żeński, precyzyjny

Adapter DIP40-1

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP40
- CON3: goldpin 40x1 męski + goldpin 40x1 żeński, precyzyjny

Adapter DIP40-2

- CON1: goldpin 5x2 męski
- CON2: podstawka precyzyjna DIP40
- CON3: goldpin 40x1 męski + goldpin 40x1 żeński, precyzyjny

Kabel łączeniowy

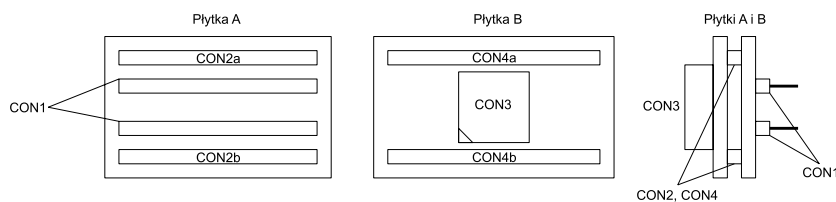
Złącze FC-10 na kabel taśmowy 2szł. Przewód taśmowy 1x10 20cm

Kabel do programowania

Złącze FC-10 Przewód taśmowy 1x10 20cm

Wykaz elementów przejściówki PLCC

- CON1: goldpin 1x40 żeński
- CON2: goldpin 1x40 żeński
- CON3: podstawka PLCC44



Rys. 24. Sposób montażu elementów na przejściówce DIP40<->PLCC44

Ten sposób programowania niesie ze sobą pewne ograniczenia, gdyż dołączane równolegle sygnały z programatora sterują również układami podłączonymi do wyprowadzeń mikrokontrolera. Jeżeli wyprowadzenia te sterują innymi układami o dużej rezystancji wejściowej, to programowanie będzie przebiegało prawidłowo. Jeżeli jednak do wyjść tych będą podłączone, na przykład diody LED, to programowanie nie będzie możliwe, gdyż programator nie będzie mógł wymusić odpowiednich stanów na tych wyprowadzeniach. Kolejne ograniczenie wystąpi, gdy wyprowadzenie służące do programowania będzie połączone z wyjściem innego układu. W tym przypadku może nawet dojść do jego uszkodzenia po podaniu sygnału z programatora. Ostatnim ograniczeniem jest sygnał zerowania mikrokontrolera !MCLR. W czasie programowania na to wyprowadzenie podawane jest napięcie 12 V, które może uszkodzić dołączone do tego wejścia układy. Zewnętrzny układ zerowania mikrokontrolera w postaci rezystora i kondensatora również ograniczy wartość napięcia programującego, uniemożliwiając programowanie. Aby umożliwić bezkolizyjne programowanie, należy na etapie projektowania uwzględnić wymagania programatora: wyprowadzenia RB3, RB6 i RB7 wykorzy-

stać jako wejściowe lub wyjściowe, ale sterujące obciążeniami o dużej rezystancji wejściowej (na przykład wyświetlacz LCD), do zerowania mikrokontrolera przy włączeniu zasilania najlepiej wykorzystać wewnętrzny moduł zerowania, a jeśli nie jest to możliwe, to zewnętrzny zamontować dopiero po ostatecznym zaprogramowaniu mikrokontrolera.

Jak widać, taka metoda programowania jest dość kłopotliwa, jeśli jednak nie ma jeszcze gotowego programu, to jest to jedyny sposób sprawdzenia nowego programu. Wykonanie kabla sprowadza się do montażu złącza FC-10 na odcinku przewodu taśmowego, analogicznie jak dla kabla do podstawek programujących, jednak złącze należy zamontować z jednej strony kabla. Drugą stronę należy rozszyć i odpowiednio oznaczyć sygnały: CLK, DATA, PGM, VPP, GND. Tak wykonany kabel należy przylutować bezpośrednio do mikrokontrolera (do odpowiednich nóżek, w zależności od układu), a drugą stroną połączyć z programatorem. Sygnały należy połączyć następująco: CLK-> RB6(GP1), DATA->RB7(GP0), PGM->PGM, VPP->!MCLR/VPP, GND->GND. Po ostatecznym zaprogramowaniu mikrokontrolera kabel należy odłączyć od płytki układu i zamontować ewentualny układ zerowania.

Podstawka-przejściówka do programowania układów w obudowie PLCC

Przejściówka pozwala na zaprogramowanie w podstawce programatora układu umieszczonego w obudowie PLCC44, zarówno PIC16, jak i PIC18. Schemat tej przejściówki jest przedstawiony na rys. 23. Przejściówka składa się z dwóch płytek połączonych ze sobą. Płytki A zawiera złącza CON1 i CON2. Złącze CON1 jest wykonane w postaci złącza szpilkowego Goldpin i służy do połączenia płytki z podstawką programatora. Druga płytka zawiera złącza CON3 i CON4. Złącze CON3 jest podstawką PLCC44 służącą do zamontowania programowanego układu. Natomiast złącze CON4 służy do połączenia z płytką A, i jest to to samo złącze, co CON2 na płytce A.

Sposób połączenia obu płytek jest przedstawiony na rys. 24. Montaż elementów należy rozpocząć od wlutowania złącza CON1 na płytce A oraz złącza CON3 na płytce B. Następnie należy połączyć obie płytki złączem szpilkowym, dla płytki A jest to złącze CON2, a dla płytki B złącze CON4. Po zmontowaniu płytek można przejść do programowania układu, wkładając do złącza CON3 mikrokontroler, a złącze CON1 do podstawki programatora. Proces programowania przebiega analogicznie, jak przy programowaniu układów w obudowach typu DIP.

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

*Wzory płytek drukowanych w formie PDF są dostępne w Internecie pod adresem: **pcb.ep.com.pl** oraz na płycie CD-EP7/2004B w katalogu PCB.*