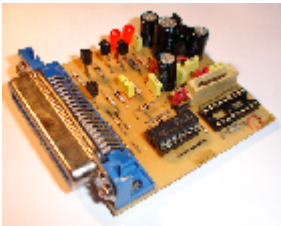


Programator mikrokontrolerów PIC

Marek SP9XUH
www.sp9xuh.pl
poczta@sp9xuh.pl



Potrzeba wyposażenia się w programator mikrokontrolerów PIC wystąpiła w momencie budowy elektronicznego klucza telegraficznego, opartego na procesorze PIC16F84. Programatory oferowane przez firmy są dobre, profesjonalnie wykonane, ale jednocześnie drogie. Podstawowy programator jest prosty w budowie, niewiele kosztuje i można bez problemu zbudować go samemu. Pierwszą decyzją jaką musiałem podjąć to wybór portu do którego będzie podłączony programator: USB, LPT czy COM. Współczesne komputery wyposażone są najczęściej tylko w porty USB, dlatego ich zastosowanie byłoby najbardziej korzystne. Jednak większość programatorów USB oparta jest na mikrokontrolerze, który i tak musimy wcześniej zaprogramować. Ostatecznie postanowiłem wykorzystać port LPT, w moim starszej generacji laptopie TOSHIBA.

Programator pozwala na wykonywanie wszystkich potrzebnych operacji na pamięci dołączonego mikrokontrolera takich jak zapis, odczyt, kasowanie i weryfikację. Jego konstrukcja umożliwia pracę w trybie ICSP (In Circuit Serial Programming), co oznacza możliwość programowania naszego procesora wprost w układzie. Klucz napięciowy zbudowany jest w oparciu o notę aplikacyjną AN589 firmy Microchip. Dokumentacja przedstawia zasady, według których procesor jest programowany. Między innymi dowiadujemy się, że procesor wymaga podwyższonego napięcia Resetu (MCLR/), do rozpoczęcia programowania.

Opisany układ spełnia funkcję programowania w obu trybach:

- HVP (High Voltage Programming),
- LVP (Low Voltage Programming).

Programator posiada podstawkę w której możliwe jest programowanie procesorów w obudowach DIP18, DIP14. Wspomniane wcześniej złącze ICSP, pozwala programować mikrokontroler bezpośrednio w układzie w którym pracuje. Można także do tego złącza podłączyć płytkę z podstawkami pod inne typy mikrokontrolerów np. 16F876 (podstawka DIP28).

Schemat ideowy

Jak widać na schemacie poniżej, w programatorze możemy wyróżnić dwa bloki;

- blok zasilania,
- blok funkcjonalny.

Zasilacz dostarcza napięcie, +5V do zasilania mikrokontrolera oraz około +13V dla programowania w trybie HVP. Napięcie +5V otrzymujemy z układu LM7805L, pracującego w typowej dla siebie konfiguracji (in, gnd, out, kondensatory filtrujące; C8 elektrolityczny na wejściu, C9 ceramiczny na wyjściu)

Napięcie resetu powinno wynosić około 13,2V i musi być stabilizowane. Najłatwiej byłoby zastosować scaloną przetwornicę wytwarzającą potrzebną wartość napięcia. Ja użyłem typowego stabilizatora LM7812L, przesuwając punkt odniesienia GND o 1,2V. Realizują to dwie diody (D2, D3) spolaryzowane w kierunku przewodzenia. Mostek Gretna umożliwia zasilanie programatora także ze źródła napięcia zmiennego. Aby zapewnić prawidłową pracę stabilizatora, wartość napięcia na kondensatorze C4 powinna wynosić nie mniej jak 16V (zależy od stabilizatora). Diody D4 i D5 sygnalizują istnienie napięć zasilania.

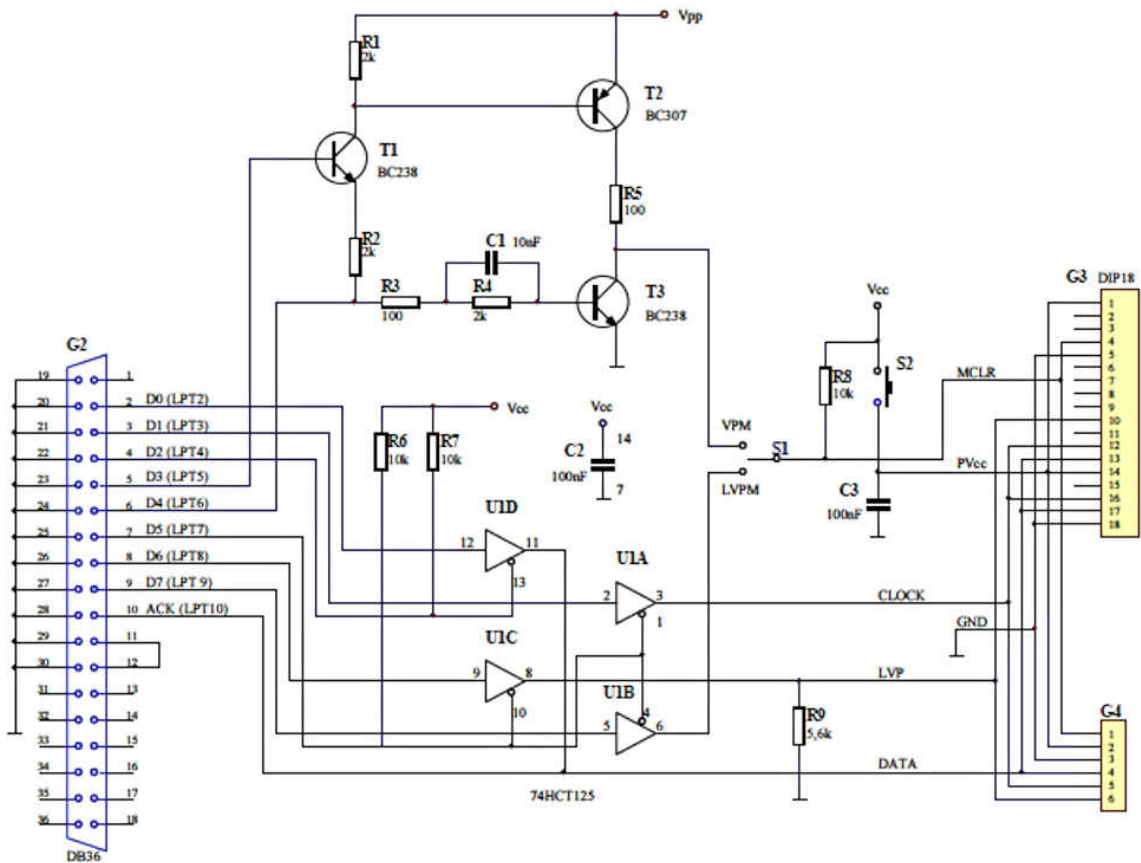
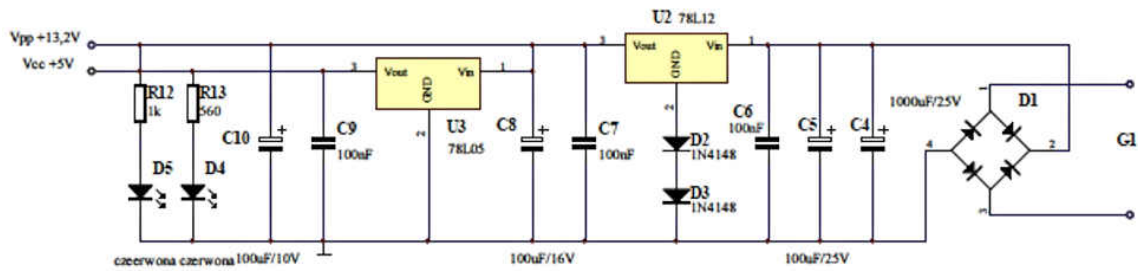
Uwaga: Maksymalny prąd dla stabilizatorów 78xxL wynosi 100 mA. Przy programowaniu mikrokontrolera w układzie i zasilania go z programatora należy pamiętać o tym, aby tej wartości prądu nie przekroczyć. Oczywiście w miejsce układów 78xxL można zastosować 78xx, których wydajność prądowa jest dużo większa; około 1,5A.

Blok funkcjonalny to:

- klucz tranzystorowy (T1, T2, T3), podający impulsy na wejście Reset (MCLR/), kiedy programujemy w trybie HVP (przełącznik S1 w pozycji HVP),
- cztery trójstanowe bufor, znajdujące się w strukturze układu scalonego 74HCT125

Stosowanie buforów wykonanych technologią HCT jest zalecane ze względu na to, że na wyjściach bramek otrzymujemy pełny zakres napięcia zasilania. Rezystory R6 i R7 utrzymują wysoki stan na wejściu bramek, gdy odłączone jest sterowanie poprzez port LPT. Podobnie rezystor R9 utrzymuje poziom niski na wejściu LVP procesora, kiedy programator jest odłączony.

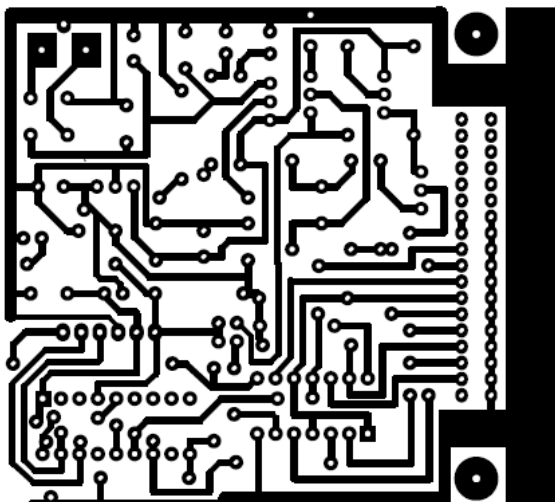
Przełącznik S1 służy do wyboru trybu programowania (HVP/LVP). Jeżeli programowany mikrokontroler jest zasilany z układu w którym pracuje, wtedy zworę S2 pozostawiamy rozwartą.



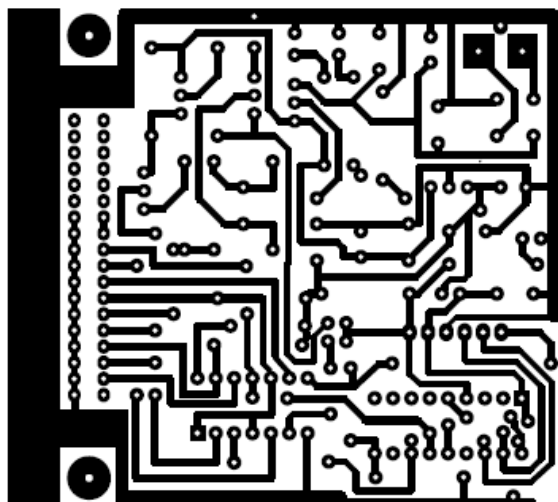
Płytką drukowana

Płytkę zaprojektowałem jako jednostronną. Wykonałem ją "technologią" termotransferu, według mnie najprostszą i najtańszą dla amatora. Jednak ze względu na dosyć duże pola lutownicze i szerokie ścieżki nie obyło się bez kilku zwór

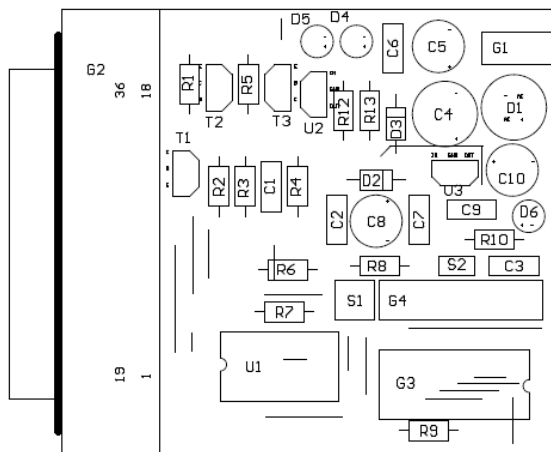
Mozaika od strony druku



Lustrzane odbicie mozaiki



Rozmieszczenie elementów



Układ jest na tyle prosty, że jego montaż nie sprawia kłopotu. Należy tylko pamiętać o krosach które trzeba wykonać przed montażem podstawki pod mikrokontroler, bo później może być problem. Po sprawdzeniu prawidłowości montażu i usunięciu ewentualnych zwarcí, podłączamy napięcie zasilania, ok. 16V. Sprawdzamy wartość napięć +5V i +13V. Jeżeli jest wszystko w porządku wkładamy w podstawkę układ scalony U1 i programator jest gotowy do działania.

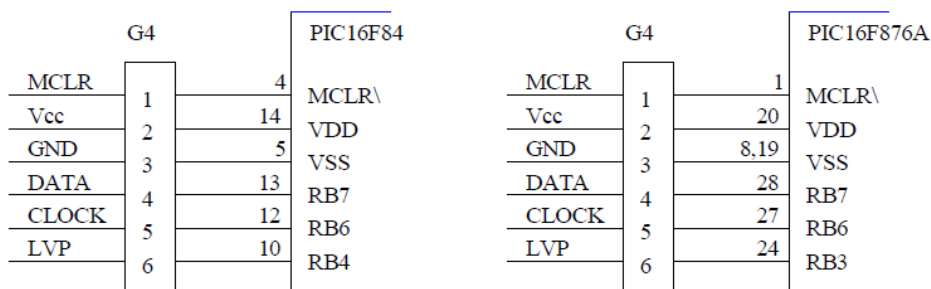
Programowanie

Pierwszą czynność jaką musimy wykonać, to prawidłowo ustawić przełącznik trybu programowania S1 (zworka) na pozycję HVP (High Voltage Programming) lub LVP (Low Voltage Programming). O sposobie programowania danego typu mikrokontrolera dowiemy się z jego dokumentacji, którą bez problemu można pobrać ze strony Microchip-a. Dla przykładu procesory z serii 16F8xx możemy programować w trybie LVP.

Złącze ICSP w które wyposażony jest programator, umożliwia programowanie procesora bezpośrednio w układzie w którym pracuje. Według mojej oceny jest to najlepsze rozwiązanie ze względu na to, że mikrokontroler może być na stałe wlutowany w płytkę i nie trzeba wkładać i wyciągać go z podstawki, co grozi uszkodzeniem nóżek.

Opis wyprowadzeń złącza programującego ICSP			
Pin	Funkcja	Pin procesora	Opis
1	MCLR	MCLR	wejście napięcia programowania
2	Vcc	VDD	plus zasilania
3	GND	VSS	minus zasilania
4	DATA	RB7	zegar
5	CLOCK	RB6	dane
6	LVP	RB3/4/5	niskie napięcie programowania

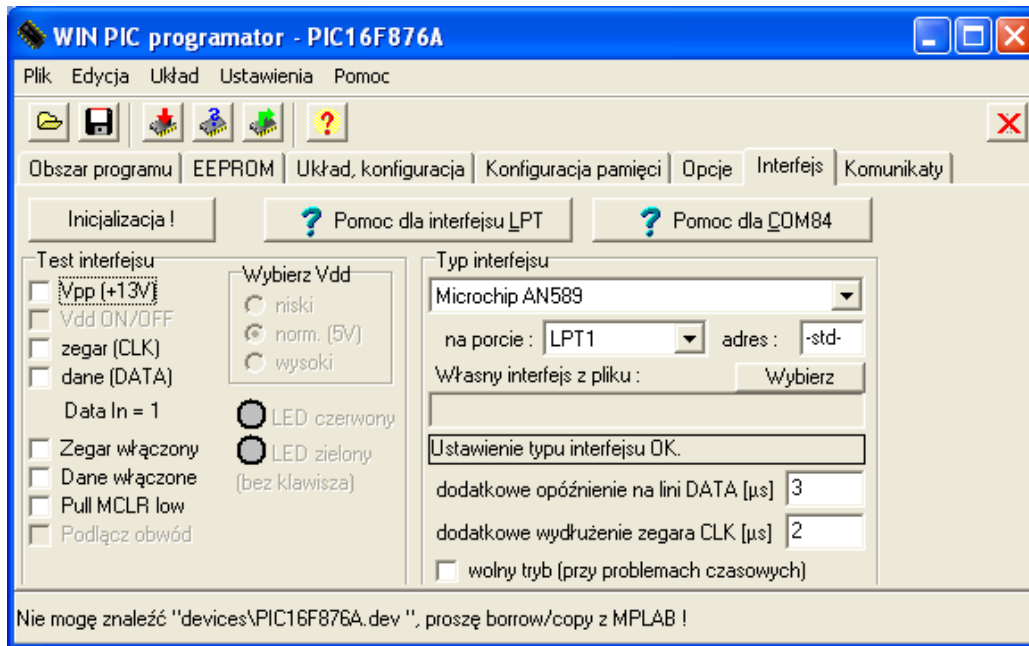
Przykład podłączenia mikrokontrolera PIC16F84 i PIC16F876A.



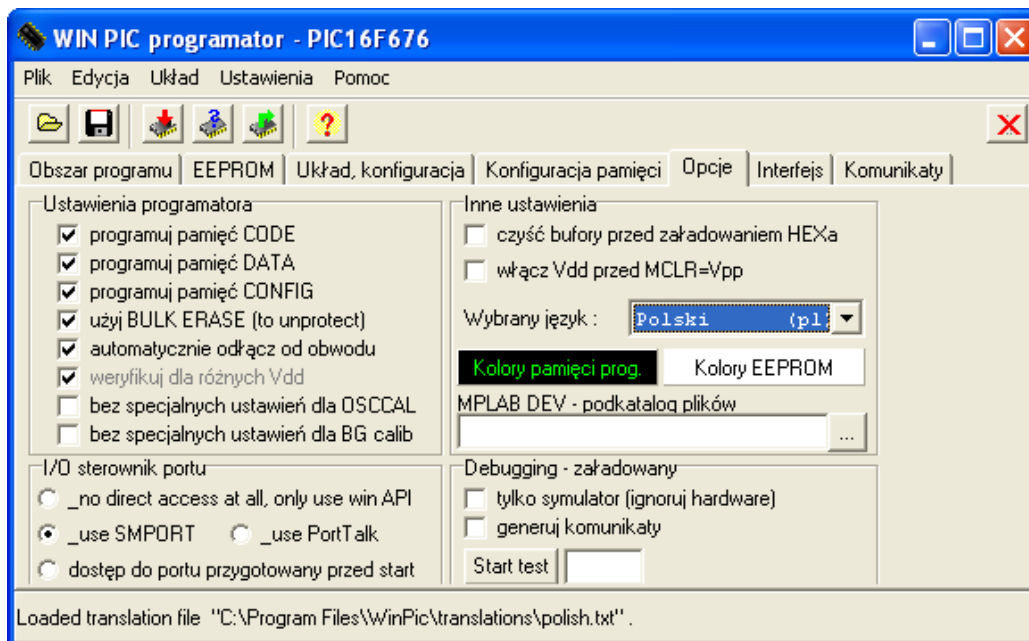
Wkładamy mikrokontroler w podstawkę lub podpinamy złącze ICSP i... potrzebujemy jeszcze oprogramowania. Programator testowałem dwoma programami, WinPic i IC-Prog. Na jednym jak i drugim wszystkie operacje na procesorze (zapis, odczyt, ustawianie bitów konfiguracyjnych,...) przebiegały bezbłędnie. Prędkość programowania nie jest może oszałamiająca, ale to tylko LPT.

WinPic

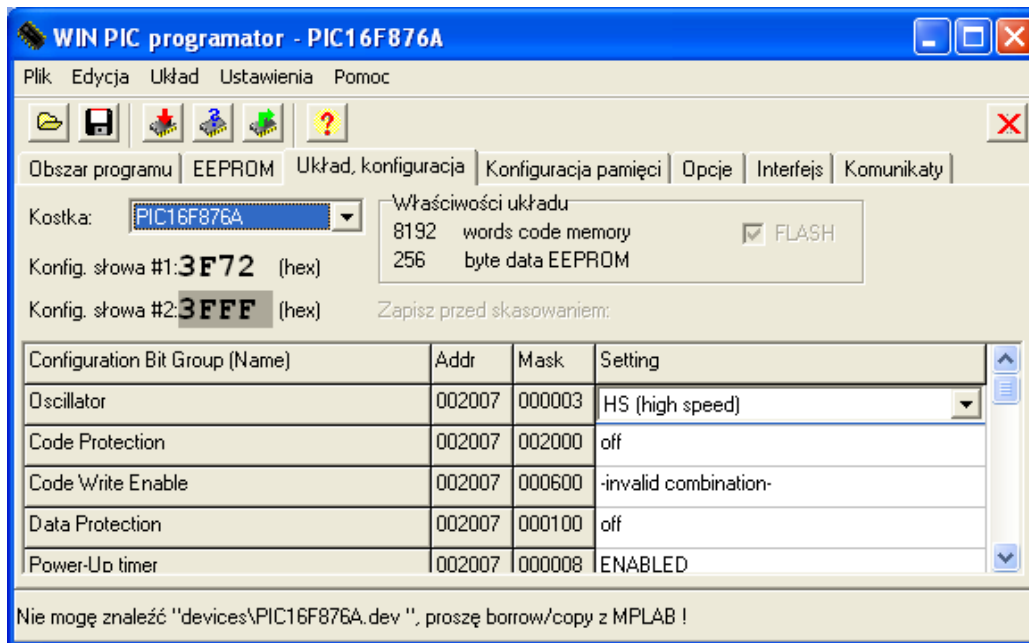
Program można pobrać download lub bezpośrednio ze strony autora <http://www.qsl.net/dl4yhfw/winpicpr.html>. Rozpakowujemy pobrany plik, instalujemy i możemy przystąpić do konfiguracji. W zasadzie pozostaje nam tylko ustawienie rodzaju interfejsu (Interface/Interfejs) na Microchip AN589 i wybranie numeru portu LPT do którego podłączony jest programator. Ewentualne błędy programowania procesora możemy próbować usunąć zmieniając parametry opóźnienia na linii DATA, wydłużenia zegara CLK lub włączyć tryb wolny.



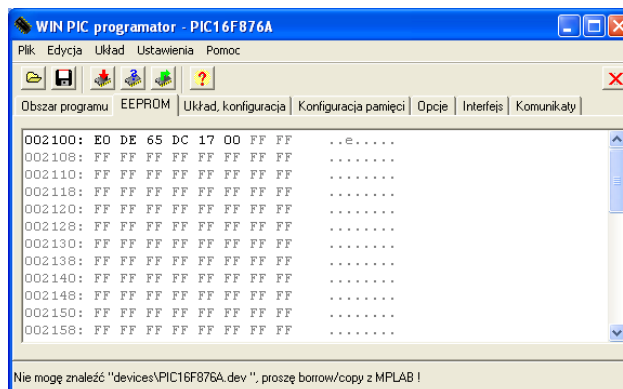
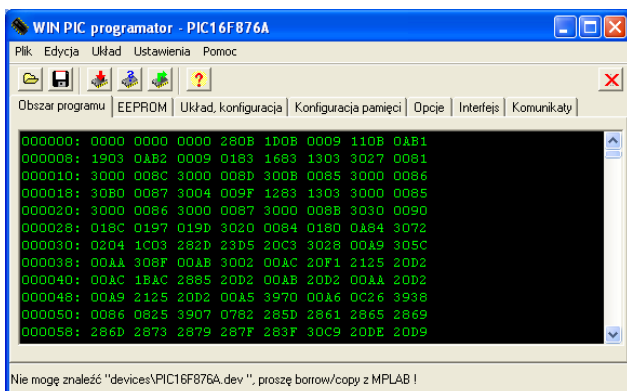
Dla ułatwienia możemy wybrać polską wersję językową (*Options/opcje*).



W zakładce *Device config/Układ konfiguracja* ustawiamy typ procesora który będziemy programować i interesujące nas bity konfiguracyjne



Po załadowaniu kodu (*File/Open, Plik/Otwórz*), który chcemy wgrać do procesora, w zakładce *Code/Obszar programu* ujrzymy dane dla pamięci Flash, a w zakładce *Data/EEPROM* te które będą umieszczone w pamięci EEPROM.



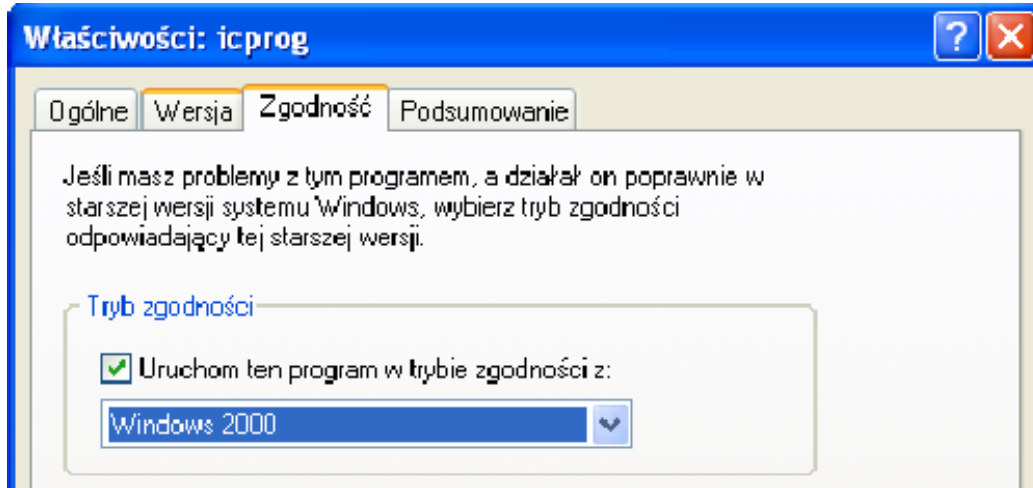
Klikamy *Device/Program(Układ/Programowanie)* i po chwili mamy zaprogramowany procesor. Jest to oczywiście tylko przykład. Warto zapoznać się z dokumentacją programu i poznać wszystkie jego możliwości.

IC-Prog

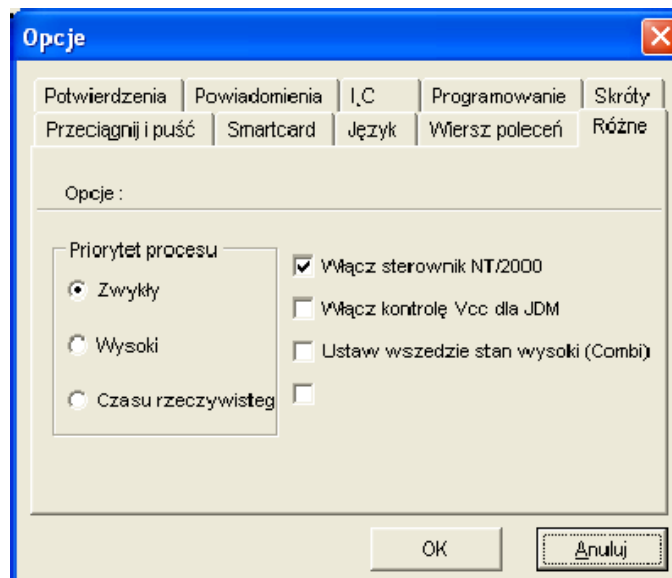
Program można pobrać w dziale Download lub ze strony programu www.ic-prog.com. Po rozpakowaniu otrzymujemy dwa pliki `icprog.exe` oraz `icprog.sys`. `icprog.exe` to plik programu, natomiast `icprog.sys` będzie nam potrzebny jeżeli naszym systemem operacyjnym jest Windows NT, 2000 lub XP.

Dla tych systemów wykonujemy następujące operacje;

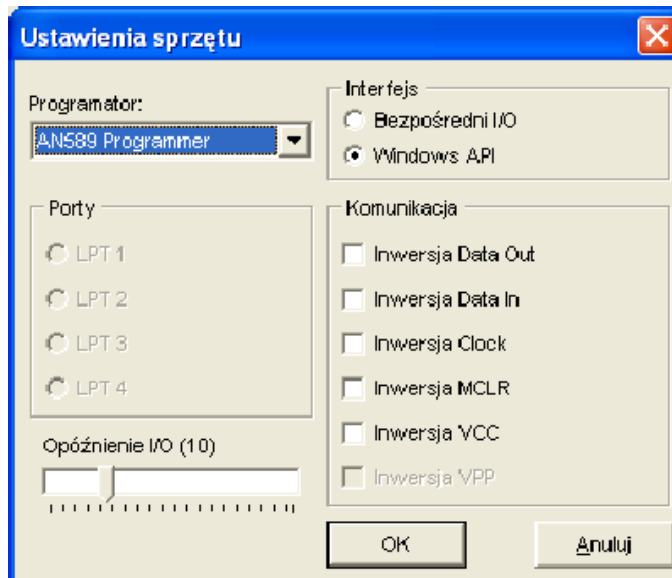
- plik `icprog.sys` kopiujemy do tego samego katalogu co `icprog.exe`,
- np. w menadżerze plików, na pliku `icprog.exe` klikamy prawym klawiszem myszy. Wybieramy *Właściwości* następnie zakładkę *Zgodność* i ustawiamy tryb zgodności jak poniżej. Klikamy *Zastosuj* i *OK*



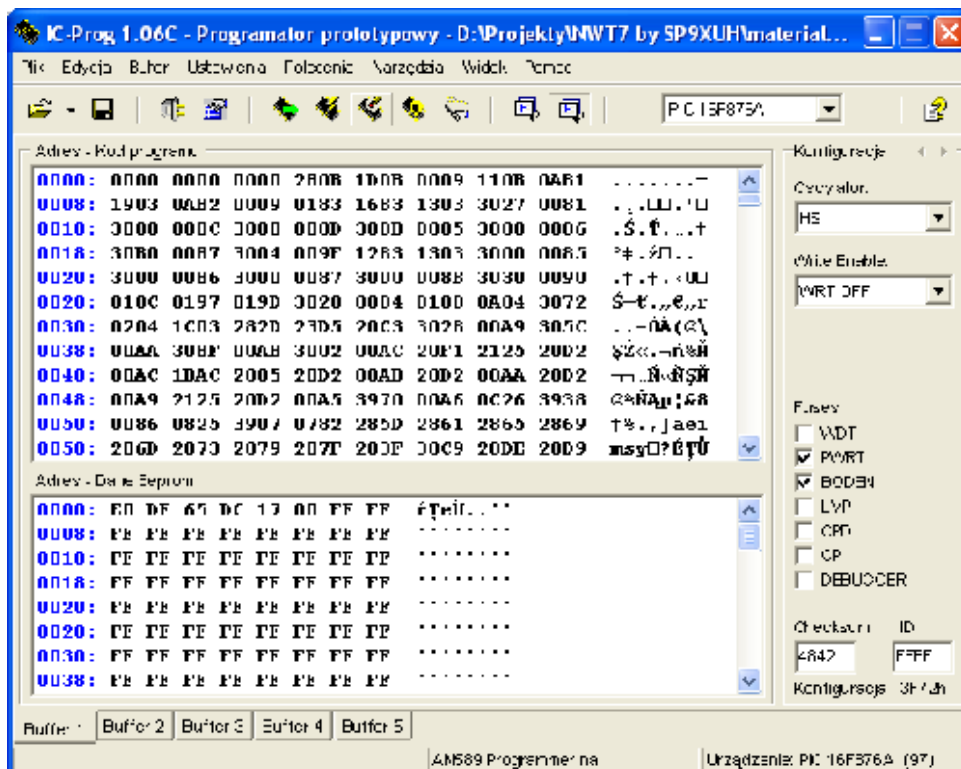
Następnie uruchamiamy ICProg, wchodzimy w *Ustawienia(Settings)* -> *Opcje(Options)* i wybieramy zakładkę *Różne(Misc)*. Zaznaczamy *Włącz sterownik 200/NT (Enable NT/2000 Driver)* i akceptujemy zmiany. Dla łatwiejszego poruszania się po programie możemy wybrać polską wersję językową.



Teraz w programie ustawiamy typ programatora. Wybieramy *Ustawienia* -> *Sprzęt* a w polu *Programator* - AN589Programmer. Dla systemu Windows 95/98 w polu *Interfejs* zaznaczamy "Bezpośredni I/O", a dla Windows NT/2000/XP, wybieramy "Windows API".



Proces programowania jest podobny jak w programie WinPic. Wybieramy typ układu - *Ustawienia* -> *Urządzenie*, ładujemy plik z kodem, ustawiamy rodzaj oscylatora i ewentualnie bity konfiguracyjne.



Podłączamy programator do układu lub wkładamy procesor w podstawkę. Z menu *Polecenia* wybieramy *Programuj wszystko* i po chwili układ jest zaprogramowany.

Pliki do pobrania:

- | | |
|-------------------------|--------------------------------|
| pic_prog_sch.pdf | schemat ideowy |
| pic_prog_bottom_pcb.pdf | plytka drukowana |
| pic_prog_top_pcb.pdf | odbicie lustrzane płytki |
| pic_prog_elementy.pdf | rozmieszczenie elementów |
| icprog106C_setup.zip | program IC-Prog wersja 1.06C |
| winpig_setup.zip | program WinPic |
| icprog_help_eng.zip | plik pomocy do IC-Prog |
| pic_prog_artykul.pdf | opis w formacie Acrobat Reader |
| pic_programator.zip | wszystkie pliki |