

# Programator USB-PROG



## Instrukcja Użytkownika

## *Spis treści*

<b>1. WPROWADZENIE.....</b>	<b>3</b>
<b>2. INSTALOWANIE STEROWNIKÓW .....</b>	<b>3</b>
<b>3. URUCHOMIENIE PROGRAMATORA .....</b>	<b>6</b>
<b>4. GNIAZDA I DIODY LED .....</b>	<b>6</b>
<b>5. POŁĄCZENIE Z PROGRAMOWANYM UKŁADEM.....</b>	<b>6</b>
5.1 WYPROWADZENIA ZŁĄCZA RJ-45 .....	7
5.2 FUNKCJE I NUMERY WYPROWADZEŃ PROGRAMOWANYCH UKŁADÓW .....	8
<b>6. WYBÓR OBSŁUGIWANEGO UKŁADU.....</b>	<b>8</b>
<b>7. OPERACJE NA UKŁADACH.....</b>	<b>9</b>
7.1.ODCZYT.....	9
7.2.ZAPIS.....	9
7.3.WERYFIKACJA .....	10
7.4.KASOWANIE .....	10
7.5.BLANK CHECK .....	10
7.6.ZABEZPIECZENIE UKŁADU .....	10
7.7.USTAWIANIE FUSE-BIT .....	10
<b>8. UWAGI ODNOŚNIE PROGRAMOWANIA ISP .....</b>	<b>11</b>
<b>9. PRZEGLĄD OBSŁUGIWANYCH UKŁADÓW.....</b>	<b>12</b>
9.1.EEPROMY SZEREGOWE 24CXX, 93CXX, 93SXX .....	12
9.2.MIKROKONTROLERY ATMEL 40-PIN.....	12
9.3.MIKROKONTROLERY ATMEL AVR.....	12
9.4.MIKROKONTROLERY MICROCHIP PIC .....	12
9.5.PAMIĘCI EEPROM – SPI.....	13
<b>BIBLIOGRAFIA .....</b>	<b>14</b>
<b>WYJAŚNIENIA.....</b>	<b>14</b>

## 1. Wprowadzenie

Programator USB-PROG, zwany dalej w skrócie „programator”, jest urządzeniem współpracującym z komputerem osobistym i służącym do programowania różnych modeli nowoczesnych mikrokontrolerów oraz szeregowych pamięci EEPROM. Łączy się z komputerem przez port USB, z którego czerpie również zasilanie. Wyposażony jest w gniazdo 8-pinowe RJ-45 (znane z kart sieciowych) do podłączenia kabla programującego. Współpracuje z oprogramowaniem COMBO2007 pod Windows 95, 98, SE, Me, NT, 2000, XP.

## 2. Instalowanie sterowników

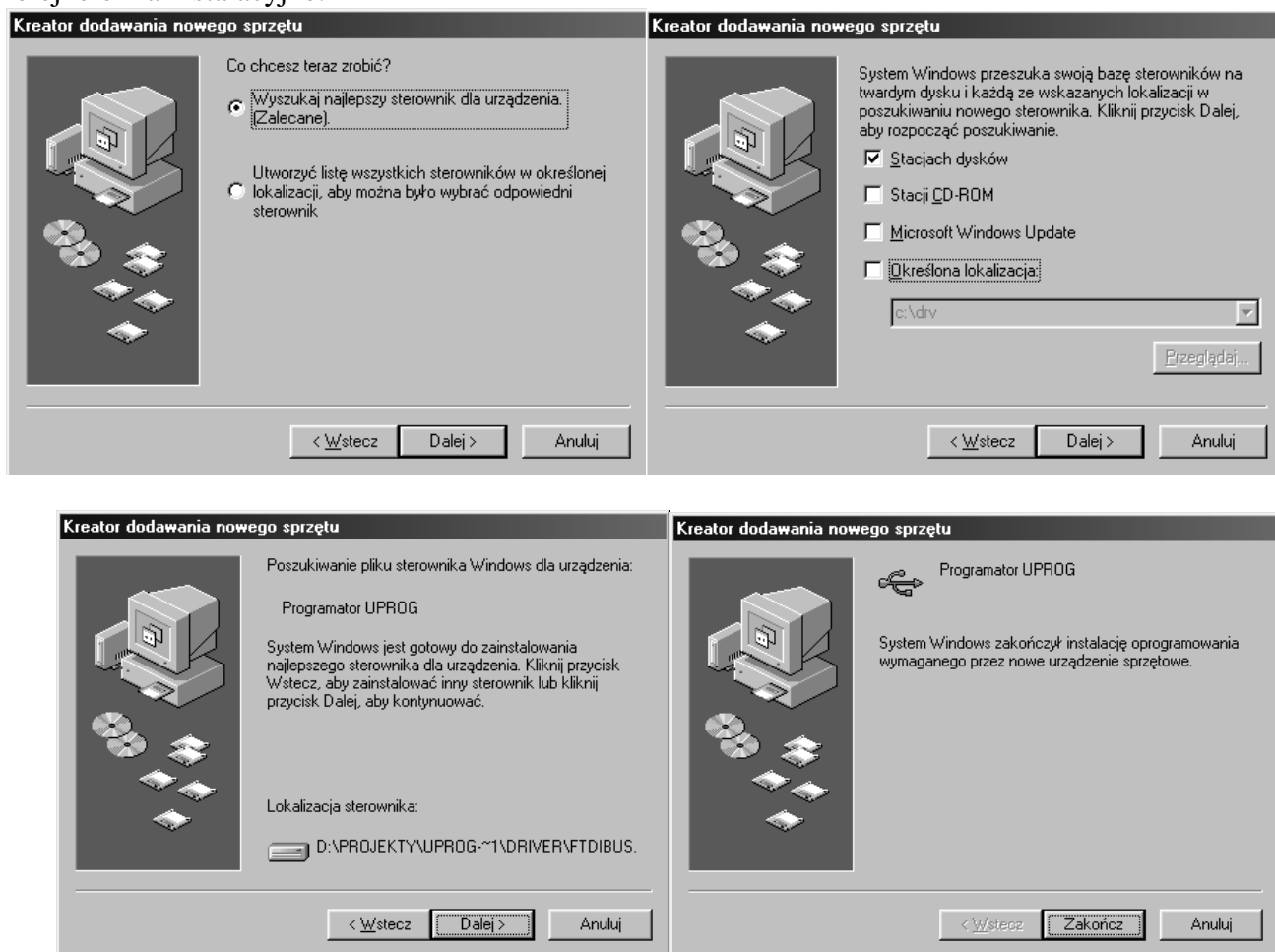
Przed rozpoczęciem pracy z programatorem należy zainstalować sterowniki programatora USB oraz oprogramowanie COMBO2007, dostarczone na dyskietce razem z programatorem. Zakładamy, że jest to pierwsze uruchomienie programatora i że sterowniki USB-PROG nie były wcześniej instalowane. W przeciwnym wypadku należy je odinstalować przed ponowną instalacją.

W pierwszej kolejności należy podłączyć USB-PROG do wolnego portu USB w komputerze. Jeżeli PC jest prawidłowo skonfigurowany, powinien natychmiast rozpoznać nowe urządzenie :



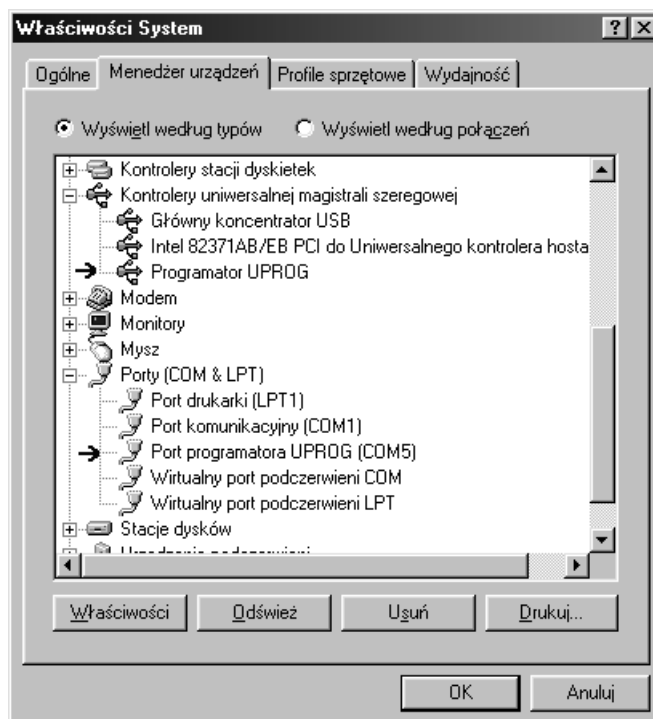
Jeżeli po podłączeniu USB-PROG-a do PC to okno się nie pojawia, oznacza to że sterowniki programatora były wcześniej zainstalowane lub USB w komputerze nie działa poprawnie i należy szukać przyczyny w konfiguracji komputera.

Następnie należy włożyć dyskietkę ze sterownikami USB-PROG do stacji dyskietek i przejść przez kolejne okna instalacyjne:



Na koniec system wykrywa jeszcze urządzenie „Port programatora UPROG” i instaluje je bez interwencji Użytkownika. W przypadku, gdyby system ponownie zapytał o sterowniki, należy wprowadzić ścieżkę dostępu (do stacji dyskietek).

Jeżeli sterowniki są zainstalowane poprawnie, w *Menedżerze urządzeń* powinny pojawić się wpisy zaznaczone strzałkami (Programator UPROG, Port programatora UPROG):



System nie powinien domagać się restartu po instalacji sterowników.

Port programatora UPROG, w zależności od konfiguracji komputera, może przyjmować dowolną wartość z zakresu COM1...COM256 i należy posługiwać się tym numerem COM podczas uruchamiania programu COMBO2007 - w przypadku błędu automatycznego wyszukiwania programatora.

Po zainstalowaniu sterowników, należy zainstalować również oprogramowanie COMBO2007 z dostarczonej dyskietki (Start-> Uruchom -> A:\setup.exe). Przy pierwszym uruchomieniu COMBO2007 może być potrzebne podanie numeru portu COM programatora.

W przypadku potrzeby odinstalowania sterowników USB-PROG, należy uruchomić z dyskietki program FTDUNIN.EXE (Start-> Uruchom -> A:\ftdunin.exe).

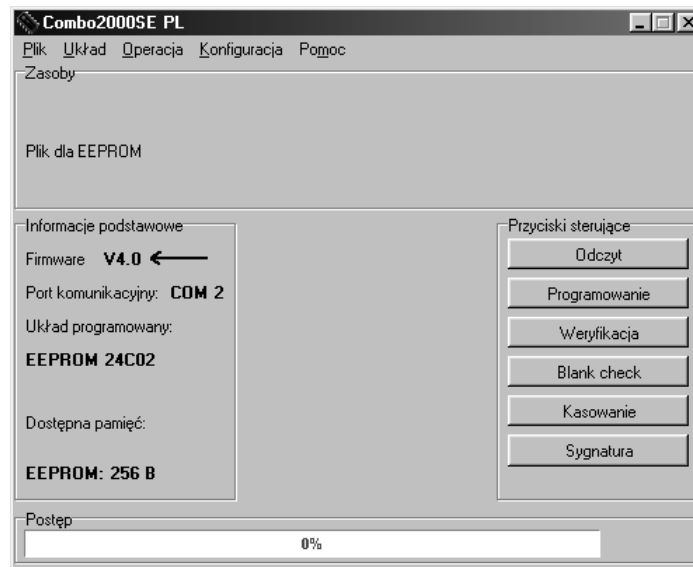
### **Uwaga !**

System Windows XP może zgłaszać podczas instalacji brak certyfikatu zgodności sterowników z systemem Windows – jest to normalny objaw, ponieważ sterowniki nie były certyfikowane przez firmę Microsoft i nie posiadają logo Windows. Należy wtedy wybrać opcję „mimo to kontynuuj”.

W niektórych instalacjach systemów Windows 2000, nowy port COM programatora USB-PROG nie będzie pojawiał się na liście programu COMBO2007 aż do momentu restartu systemu.

### 3. Uruchomienie programatora

Podczas podłączania programatora do USB, powinna zapalić się zielona LED Z i mignąć czerwona LED P, co sygnalizuje prawidłową pracę urządzenia. Po podłączeniu do portu USB, nie wymaga dalszych czynności. Wystarczy tylko uruchomić program COMBO2007 (ewentualnie wprowadzić nr COM w menu *Konfiguracja* ). Po uruchomieniu programu w okienku wyświetla się wersja Firmware :



Program COMBO2007 współpracuje zarówno z programatorem COMBO3, jak i USB-PROG. Wersje firmware 2.x, 3.x są zarezerwowane dla COMBO3, podczas gdy 4.x – dla programatora USB-PROG. Jeżeli zamiast numeru wersji wyświetla się ????, to oznacza, że sterowniki nie są zainstalowane, podano zły numer portu COM lub programator nie jest podłączony.

### 4. Gniazda i diody LED

Programator USB-PROG posiada 3 kontrolne diody LED:

- zielona : Zasilanie – sygnalizuje zasilanie urządzenia z USB,
- czerwona : Praca – sygnalizuje wykonywanie operacji na układzie,
- żółta : Dane – sygnalizuje przesyłanie danych przez USB.

Do podłączenia programowanego układu służy :

- 8-pinowe gniazdo RJ-45

### 5. Połączenie z programowanym układem

Programator USB-PROG jest dedykowany do programowania układów „w systemie” (ISP, SPI), to znaczy układów wlutowanych w płytkę drukowaną jakiegoś urządzenia i połączonych z programatorem za pomocą kabla. Programowanie ISP ma wiele zalet w porównaniu z tradycyjnym, równoległym programowaniem, i to we wszystkich fazach projektu : rozwój oprogramowania, produkcja, serwis.

W **fazie rozwoju**, dzięki ISP nie jest wymagane kosztowne urządzenie emulatora – mikrokontroler użyty do testów jest tym samym co mikrokontroler docelowy. Kość może być wlutowana na płytke docelową, a tam przeprogramowywane wielokrotnie bez potrzeby wylutowywania, a nawet bez wyłączenia urządzenia ! Nie jest wymagana specjalna podstawka testowa ani układ dopasowujący, dzięki czemu można testować również programy krytyczne czasowo.

W **fazie produkcji**, płytki urządzeń mogą być kompletnie zmontowane jeszcze przed dokończeniem oprogramowania, aby produkt mógł szybciej zaistnieć na rynku. Dodatkowe informacje, jak np. numer seryjny czy nazwa producenta, mogą być zapisane w układzie przed samym dostarczeniem do klienta. Nie ma ryzyka zalegania w magazynach nieaktualnych zaprogramowanych kości, ponieważ program może być uaktualniony w dowolnym momencie.

Nawet kiedy produkt trafi do użytkownika końcowego, może być szybko i sprawnie **uaktualniony** lub poprawiony przez serwisantów. Klienci mogą również przeprogramowywać urządzenia samodzielnie, jeśli posiadają odpowiedni sprzęt.

**Sprzęt** do programowania ISP jest relatywnie tani i łatwy w użyciu – wymaga max. 6 linii łączących programowaną kość z programatorem.

**Przed programowaniem należy połączyć wyprowadzenia kabla ISP z odpowiednimi końcówkami układu, według tabeli.** W przypadku, gdy konieczne jest programowanie nowej kości, która nie jest wlutowana do płytki, można posłużyć się **dotatkową płytką adaptera**, z podstawką na kość i kablem do połączenia z programatorem.

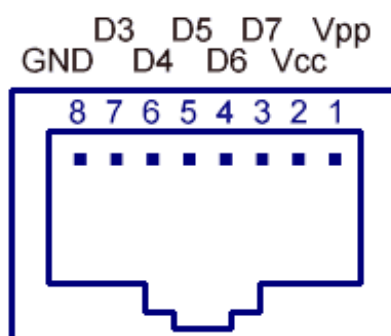
## 5.1 Wyprowadzenia złącza RJ-45

Sygnał	Wyprowadzenie układu programowanego									
	Numer w gnieździe	RJ-45	24Cxx	25Cxx	93Cxx	SDE2506	PIC	Atmel-SPI	AT45DB0x1	90S2343(HV)
Vpp	1	--	--	--	--	--	MCLR	RST <sup>2</sup>	--	RESET
Vcc <sup>1</sup>	2	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	-- <sup>4</sup>	Vcc
D7	3	SCL	SCK	CLK	CLK	B6/GP1(clk)	SCK	SCK	SCK	SCI
D6	4	SDA	SO	DO,DI	D	B7/GP0(dat)	MISO	SO	SO	SDO
D5	5	--	SI	CS	CE	--	MOSI	SI	SI	SII
GND	8	GND	GND	GND	GND	GND	GND	GND	GND	GND
D4	6	--	CS	ORG <sup>1</sup>	--	--	--	CS	CS	SDI
D3	7	--	--	WP <sup>3</sup>	--	--	--	--	--	--

### Uwagi:

- 1) Podłączane opcjonalnie, gdy programowany układ nie ma własnego zasilania i pobór prądu zasilania nie przekracza 50 mA
- 2) Linia RESET dla Atmeli rodziny 51 (tylko SPI) wymaga zabezpieczającej diody Zenera 4,7-5,6 V
- 3) Linia WP istotna tylko dla rodziny 93Sxx
- 4) Wyprowadzenia Vcc, RESET, WP układów DataFlash należy podłączyć do zewnętrznego źródła napięcia 3,3 V

Numeracja wyprowadzeń złącza RJ-45 (widok na gniazdo programatora):



## 5.2 Funkcje i numery wyprowadzeń programowanych układów

24Cxx		25Cxx		93Cxx		SDE2506		90s2343, ATtiny (HV)	
1,2,3,4,7	GND	1	CS	1	CS	1,7	GND	1	RESET
5	SDA	2	SO	2	CLK	2	CE	2	SCI
6	SCL	3,7,8	VCC	3,4	DO,DI	3	WP	4	GND
8	VCC	4	GND	5	GND	4	D	5	SDI
		5	SI	6	ORG	5	CLK	6	SII
		6	SCK	7	WP			7	SDO
				8	VCC			8	VCC

Mikrokontrolery Microchip PIC							
PIN gniazda RJ-45	Sygnał PIC	Końcówka układu programowanego					
		PIC10F20x	PIC12C50x	PIC16C505	PIC16F84	PIC16F873/6	PIC16F874/7
		6-pin	8-pin	14-pin	18-pin	28-pin	40-pin
1	MCLR	6	4	4	4	1	1
2	VDD	5	1	1	14	20	11,32
3	RB6/GP1	3	6	12	12	27	39
4	RB7/GP0	1	7	13	13	28	40
8	VSS	2	8	14	5	8,19	12,31

## 6. Wybór obsługiwanego układu

Po połączeniu z programowanym układem, należy wybrać jego typ w programie obsługi. Dokonuje się tego za pośrednictwem okienek. Niektóre rodziny układów posiadają mechanizm pozwalający wykryć ich obecność **automatycznie**. Układy te to wszystkie procesory Atmel AVR oraz pamięci EEPROM 24Cxx (z wyjątkiem 24C32..24C256, które są rozpoznawane jako 24C02). Służy do tego przycisk „**sygnatura**”. Jednak większość obsługiwanych kości (w tym AT89S8252) wymaga ręcznego wybrania w programie. Czasami zdarza się, że układ AVR nie jest rozpoznawany automatycznie mimo prawidłowego połączenia, wtedy również musi być wybrany ręcznie.

Niektóre z obsługiwanych układów nie są jawnie obecne na liście w programie i należy wybrać ich zamiennik:

Układ	Zamiennik
PIC12C508A, 12CE518	PIC12C508
PIC12C509A, 12CE519	PIC12C509
SDA2516, SDA2526, PCF8582	24C02
PIC16C62, C64, CE625, C716	PIC16C622
PIC16C63, 16C65	PIC16C73
PIC16C67, 1C76, C77	PIC16C66
SDA2546, PCF8594	24C04
SDA2586, PCF8598	24C08
93C14	93C46 (16bit.)
ER5911	93C46
ST95xxx, M95xxx	25Cxxx
ATiny 11	ATiny 12
ATmega8535	ATmega8
ATmega8515	ATmega8
ATmega 64	ATmega 128 (połowa)
ATmega 603	ATmega 103 (połowa)

## 7. Operacje na układach

### 7.1. Odczyt

Powoduje odczytanie zawartości pamięci układu do pamięci RAM komputera i zapisanie jej na dysku w pliku hex o podanej nazwie. Przed odczytem program prosi o podanie odpowiedniej nazwy pliku. W przypadku układów zawierających 2 typy pamięci : FLASH i EEPROM (np. AVR, 89S8252) trzeba podać 2 nazwy plików, pierwszą dla pamięci FLASH, drugą dla pamięci EEPROM. W przypadku układów PIC wystarcza jeden plik, ponieważ wszystkie dane (FLASH, EEPROM, FUSE, LOCK) są zapisywane w jednym pliku.

#### Uwaga!

Wszystkie mikrokontrolery posiadają możliwość **zabezpieczenia kodu programu przed odczytem** przez osoby niepowołane (tzw. **LOCK-bit** lub **CODE PROTECTION – CP**). Zabezpieczenie to jest ustawiane celowo w celu uniemożliwienia powielania układu lub analizy kodu programu. Odczyt zabezpieczonego układu zwraca przypadkowe wartości bajtów (PIC), same bajty FF (Atmel) lub kolejne wartości 0,1,2,... (Atmel przez ISP) i nie powoduje żadnych zmian w układzie. Zabezpieczenie jest skonstruowane w ten sposób, że nie jest możliwe jego zdjęcie bez skasowania programu. W układach z pamięcią FLASH (Atmel, PIC16Fxx) kasowanie zawartości układu automatycznie zdejmuje zabezpieczenie (jednak zawartość pamięci została już wyczyszczona). W układach z pamięcią EPROM (PIC 12Cxx, 16Cxx) nawet skasowanie układu promieniami UV nie zdejmuje zabezpieczenia – taki układ jest już bezużyteczny.

### 7.2. Zapis

Programuje pamięć układu zawartością pliku, o podanej wcześniej nazwie. Układy z pamięcią **FLASH** (Atmel, PIC 16Fxx) są kasowane przed programowaniem, natomiast układy z pamięcią **EPROM** (12Cxx,

16Cxx) muszą być czyste przed programowaniem. Pamięci **EEPROM** nie wymagają wcześniejszego kasowania.

Po zapisie programu dokonywana jest weryfikacja – sprawdzenie, czy układ został zaprogramowany prawidłowo. Jeżeli przebiegła pomyślnie, w zależności od typu układu na koniec programowane są dodatkowe ustawienia, jak **LOCK**-bity i **FUSE**-bity (tylko w przypadku PIC).

Jeżeli wystąpił błąd weryfikacji, oznacza to że układ jest uszkodzony, nieprawidłowo włożony do podstawki, brakuje jakiegoś połączenia lub programator jest uszkodzony.

W połączeniu z operacją odczytu, umożliwia kopiowanie układów (jeśli nie są zabezpieczone przed odczytem).

### 7.3. Weryfikacja

Porównuje zawartość pamięci układu z zawartością wskazanego pliku. Weryfikacja nie jest możliwa, jeśli układ został zabezpieczony przed odczytem !

Jeżeli weryfikacja przebiegła prawidłowo, że zawartość układu jest dokładnie taka sama jak wskazanego pliku.

### 7.4. Kasowanie

Powoduje wyczyszczenie zawartości pamięci układu (wpisanie samych FF) i wyłączenie zabezpieczeń. Nie dotyczy układów PIC z pamięcią EPROM, które są kasowane za pomocą promieni ultrafioletowych . Dane katalogowe zalecają do tego celu światło o długości fali 2537 Å z intensywnością 12,000 μW/cm<sup>2</sup> przez czas około 20 minut (dawka promieniowania 15 W\*s/cm<sup>2</sup> ). Kasowanie nie zmienia ustawień FUSE-bit w procesorach Atmel AVR.

### 7.5. Blank check

Sprawdza, czy układ jest czysty, tzn. czy jego pamięć zawiera same FF. W przypadku procesorów Atmel zabezpieczonych przed odczytem zwraca wynik pozytywny, mimo że układ został wcześniej zaprogramowany. Wynika to ze specyfiki zabezpieczeń Atmela.

### 7.6. Zabezpieczenie układu

Powoduje wpisanie wybranego ustawienia zabezpieczenia do układu i tym samym jego zabezpieczenie. Zawartość pamięci zabezpieczonego układu jest już niedostępna na zewnątrz aż do momentu jej skasowania. Nie ma żadnej możliwości „cofnięcia” zabezpieczenia.

W przypadku układów **PIC** z pamięcią EPROM (serie 12Cxx, 16Cxx), nie ma możliwości ponownego zaprogramowania raz zabezpieczonego układu (nawet w kosztownej wersji „okienkowej” !). Dlatego należy używać zabezpieczenia ze szczególną ostrożnością.

Zabezpieczenie jest ustawiane automatycznie po zaprogramowaniu układu (jeśli wybrano poziom zabezpieczenia).

### 7.7. Ustawianie FUSE-BIT

Opcja dotyczy tylko niektórych układów **AVR** oraz wszystkich **PIC**. FUSE-bity służą do konfiguracji specyficznych funkcji sprzętowych układu, jak np. szybki start, wybór oscylatora, wyłączenie programowania ISP lub włączenie układu watchdog.

Wybranie opcji „ustaw FUSE-bit” powoduje zaprogramowanie wybranych ustawień. W układach AVR ustawienia nie są zmieniane podczas kasowania układu i aby zmienić konfigurację układu, należy uruchomić „Ustaw FUSE”.

W układach PIC należy ustawić FUSE po zaprogramowaniu i sprawdzeniu układu; razem z bitami FUSE może być zaprogramowane zabezpieczenie kodu CP.

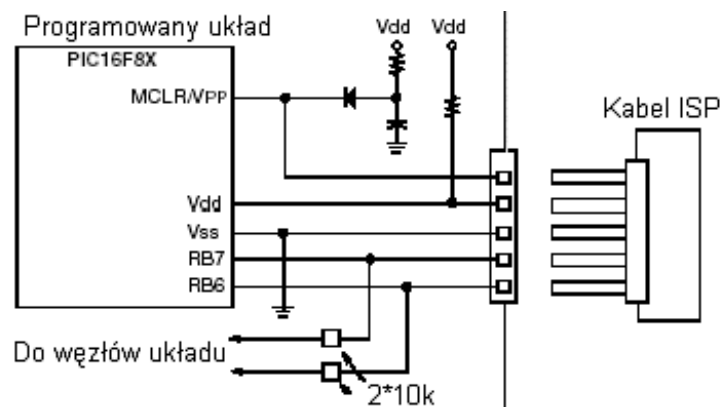
## 8. Uwagi odnośnie programowania ISP

Zalecana **długość przewodu ISP** to 20 cm. W przypadku dłuższego przewodu, do prawidłowej pracy może być konieczne połączenie linii CLK i GND kondensatorem 100pF tuż przy samym układzie. Opisywany problem wystąpił z układami PIC16F877.

**Aby programowanie ISP było możliwe, muszą być spełnione następujące warunki :**

1. Kość musi sprzętowo umożliwiać ISP.
2. Programowanie ISP musi być **włączone** w kości (FUSE **SPIEN** w AVR, 89S8252) – zazwyczaj układy dostarczane przez producenta są gotowe do ISP, lecz w niektórych aplikacjach celowo wyłącza się programowanie ISP w celu zwiększenia bezpieczeństwa układu i zapisanego programu.
3. Linie, do którego podłączono programator (np. CLK, DAT, RES, MISO, MOSI ) nie mogą być zbyt obciążone w układzie, tak aby programator mógł sterować ich poziomami. Nie mogą tu być bezpośrednio podłączone zworki, wyłączniki, duże pojemności ani wyjścia innych aktywnych elementów na płytce. Rezystor podciągający RESET nie może być mniejszy od 10 k $\Omega$ , kondensator RESET nie może być większy niż 1  $\mu$ F.

Przykład rozwiązania dla PIC16F84 przedstawia rysunek:



4. Kabel ISP nie może być zbyt długi. **Typowa długość to 20 cm**, czasem wymagany jest kondensator 100pF na sygnale CLK – patrz wyżej.
5. W niektórych układach PIC (np. 16F877) należy zwrócić szczególną uwagę na ustawienie FUSE **LVP**. Jeżeli w programie zadeklarowano **LVP\_ON**, należy zapewnić stan wysoki linii **PGM (RB3)** aby umożliwić programowanie a potem stan niski, aby umożliwić wykonywanie programu. Zaleca się deklarowanie ustawienia **LVP\_OFF**.
6. Do układu musi być doprowadzone zasilanie. Najczęściej wyjście VCC z programatora jest za słabe, aby zasilić układ programowany w systemie wraz z jego otoczeniem.
7. Układy **Atmel** wymagają obecności kwarcu 1-12 MHz lub innego źródła taktowania przy ISP tak jak przy normalnej pracy układu. Częstotliwość taktowania układu nie może być niższa od 1MHz.

## 9. Przegląd obsługiwanych układów

### 9.1. EEPROMy szeregowo 24Cxx, 93Cxx, 93Sxx

Układy **24Cxx**, **24LCxx** są programowane przez interfejs I<sup>2</sup>C. Są to typowe pamięci EEPROM o pojemności od 256 bajtów do 32 kilobajtów. Możliwe jest ich programowanie w systemie lub w podstawce – wykorzystywane linie to VCC, GND, SCL, SDA.

Układy **93Cxx** są programowane interfejsem Microwire. Pojemność od 128 bajtów do 1024 bajtów. Większość tych układów ma możliwość programowania zarówno w trybie 8-bitowym, jak i 16-bitowym (wybór dokonywany pinem 6 – ORG). Zdarzają się jednak egzemplarze programowane wyłącznie w trybie 16-bitowym (np. 93C06, 93C66). W takim przypadku należy koniecznie ustawić w programie tryb 16-bit. Mogą być programowane w systemie : linie VCC, GND, CLK, DAT, CS.

Układy **93Sxx** są odmianą 93Cxx z możliwością ustawienia zabezpieczenia przed zapisem dla całości lub części pamięci. Zabezpieczenie może być możliwe do edycji, lub może być ustawione permanentnie – wtedy układ staje się układem OTP i może być tylko odczytywany. Do ustawiania zabezpieczenia służy rejestr PROTECTION REGISTER.

### 9.2. Mikrokontrolery Atmel 40-pin

W tej grupie ISP udostępniają jedynie układy AT 89S8252, 89S51, 89S52, 89S53. Programowanie ISP może być jednak zablokowane odpowiednim FUSE-bitem w celu zwiększenia bezpieczeństwa (podczas programowania równoległego). Układy te nie są automatycznie rozpoznawane i wymagają diody zabezpieczającej Zenera na lini RESET – nie tolerują napięcia 12V.

### 9.3. Mikrokontrolery Atmel AVR

8-bitowe mikroprocesory RISC (około 100 rozkazów, 16-bitowy rdzeń). Wszystkie układy z tej serii posiadają pamięć FLASH na kod programu (1-128 kB) i EEPROM na dane (64-512 B). Dodatkowo posiadają możliwość programowania ISP (za pomocą interfejsu SPI). Spotykane są wersje 8-, 20- i 40-pinowe. Konfiguracja układu jest ustalana za pomocą 2 bitów FUSE, zależnie od układu : FSTRT – szybki start, RCEN – włączenie wewnętrznego generatora RC, SPIEN – włączenie możliwości programowania ISP. Uwaga! Wartości bitów FUSE są zanegowane : 0 oznacza „włączony” ! Oczywiście umożliwiają też zabezpieczenie pamięci przed odczytem i zapisem. Układy 8-pinowe, jak 90S2323, 90S2343, ATiny, oprócz trybu ISP, umożliwiają programowanie HV (z napięciem VPP=12V). W tym trybie można programować wszystkie FUSE bity. Układy ATmega posiadają do 1Mbit pamięci FLASH i są programowane stronicowym algorytmem.

### 9.4. Mikrokontrolery Microchip PIC

Jest to obszerna i bardzo różnorodna grupa układów. Wszystkie z nich to 8-bitowe układy **RISC** z rdzeniem 12- lub 14-bitowym, pamięcią kodu programu EPROM lub FLASH, opcjonalną pamięcią EEPROM na dane. We wszystkich układach konfiguracja ustawiana jest za pomocą kilku lub kilkunastu bitów FUSE (np. WDTE – włączenie WDT, OSC RC- włączenie oscylatora RC, LVP- umożliwienie programowania 5V, MCLR- włączenie wyprowadzenia RESET itp). Każdy typ układu ma swoje indywidualne możliwości konfiguracji i przed zaprogramowaniem należy upewnić się w opisie układu (PDF), jakie powinny być prawidłowe (dla tej aplikacji) ustawienia FUSE. Ustawienia bitów FUSE deklaruje się w pliku źródłowym asm w linii CONFIG jak np:

```
__CONFIG    _CP_OFF & _WDT_ON & _RC_OSC
```

Ustawienia FUSE, tak jak i zawartość danych dla EEPROM są przechowywane standardowo w jednym pliku, razem z kodem programu.

#### **Podstawowe podgrupy układów PIC:**

- 6-pinowe (SOT-23), np. 10F202,
- 8-pinowe, np. 12C508, 12C509A, 12CE518, 12CE677 – 12 lub 14 bitowe, z pamięcią EPROM na kod ewentualnie (12CExxx) EEPROM na dane,
- 8-pinowe z pamięcią FLASH (12F509, 12F629),
- 14-pinowy 16C505 – z pamięcią EPROM, wiele możliwych ustawień FUSE
- 18-pinowe z pamięcią FLASH : 16F84, 16F628 itp (14-bitowy rdzeń, pamięć danych EEPROM, bardzo popularne)
- 40- lub 28- pinowe z pamięcią FLASH kodu i EEPROM danych : np. 16F877, 16F873 – bardzo wydajne i funkcjonalne, bogate peryferia, wiele możliwości ustawień.

Warto tu jeszcze raz zaznaczyć różnicę pomiędzy pamięcią kodu **EPROM i FLASH**. Pamięć EPROM ( w układach 12Cxx, 16Cxx) może być zaprogramowana tylko raz ( z wyjątkiem drogich i żadkich układów w obudowach „okienkowych” ). Nawet w wersji okienkowej nie jest możliwe skasowanie zabezpieczenia kodu promieniami UV. Źle zaprogramowany układ jest już całkowicie bezużyteczny !

Natomiast pamięć **FLASH** (układy 16Fxx, 16C84) może być przeprogramowywana praktycznie dowolną ilość razy, układ taki z powodzeniem nadaje się do aplikacji testowych.

#### **9.5. Pamięci EEPROM – SPI**

Programowane są przez interfejs SPI, mogą być programowane w systemie. Pojemności od 128 bajtów do 32 kilobajtów. Obudowy 8-pinowe. W tej grupie znajduje się rodzina 25Cxx, 25LCxx oraz -przez podobieństwo algorytmu programowania – SDE 2506. Układy 25Cxx mają możliwość zabezpieczenia przed zapisem całości lub części ( $\frac{1}{2}$  ,  $\frac{1}{4}$ ) pamięci. Zabezpieczenie ustawiane jest w rejestrze STATUS REGISTER.

Układy Atmel DataFlash ® - AT 45DB011, AT 45DB081 i podobne, są pamięciami FLASH o dużej pojemności (np. 1 megabajt) z interfejsem SPI. Należy je podłączyć następująco do programatora : CS-CS(D4), SCK-SCK, SI-MOSI, SO-MISO, WP-Vcc, RESET-Vcc, GND-GND, Vcc-3,3V z zewn. zasilacza.

## Bibliografia

1. Strona domowa programatora : <http://www.progstar.com.pl/uprog>
2. Aktualne oprogramowanie pod Windows : <http://www.combosoft.republika.pl>
3. Strony katalogowe i noty aplikacyjne firmy Atmel: <http://www.atmel.com>
4. Strony katalogowe i noty aplikacyjne firmy Microchip: <http://www.microchip.com>
5. Podręcznik Microchip : In-Circuit Serial Programming (ICSP™) Guide

## Wyjaśnienia

Producent programatora USB-PROG nie ponosi żadnej odpowiedzialności za szkody wynikłe z eksploatacji urządzenia.

Kopiowanie kodu zapisanego w pamięci mikrokontrolera lub pamięci EEPROM bez zgody autora rzeczonoego kodu jest naruszeniem własności intelektualnej i jest niezgodne z obowiązującym prawem.

Producent programatora programatora USB-PROG nie ręczy za żadne szkody, które powstały przez nielegalne zastosowanie programatora USB-PROG.

Wszelkie znaki handlowe użyte w niniejszej instrukcji są własnością ich zarejestrowanych posiadaczy i zostały tu przytoczone jedynie w celu identyfikacji ich produktów.

