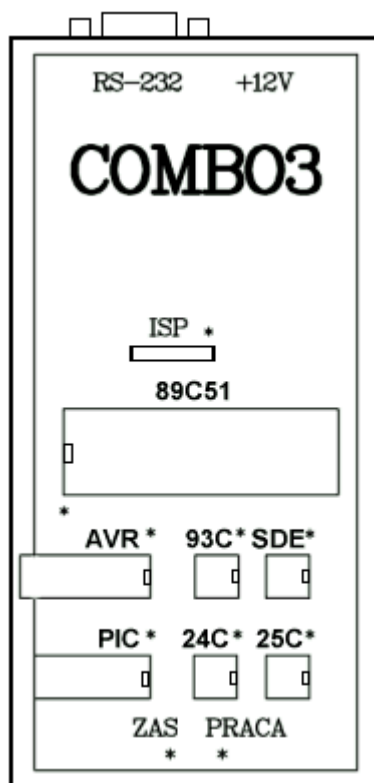




# Programator COMBO3



## Instrukcja użytkownika

Wersja 1.1

# Spis treści

<b>1. WPROWADZENIE.....</b>	<b>3</b>
<b>2. URUCHOMIENIE PROGRAMATORA.....</b>	<b>3</b>
<b>3. INSTALOWANIE OBSŁUGIWANEGO UKŁADU W PODSTAWCE. ....</b>	<b>3</b>
<b>4. WYBÓR OBSŁUGIWANEGO UKŁADU.....</b>	<b>5</b>
<b>5. OPERACJE NA UKŁADACH.....</b>	<b>4</b>
5.1. Odczyt.....	5
5.2. Zapis .....	6
5.3. Weryfikacja.....	6
5.4. Kasowanie.....	6
5.5. Blank check .....	7
5.6. Zabezpieczenie układu.....	7
5.7. Ustawianie FUSE-BIT.....	7
<b>6. PROGRAMOWANIE ISP .....</b>	<b>7</b>
<b>8. PRZEGLĄD OBSŁUGIWANYCH UKŁADÓW.....</b>	<b>10</b>
8.1. EEPROMy szeregowo 24Cxx, 93Cxx, 93Sxx.....	10
8.2. Mikrokontrolery Atmel 20-pin .....	11
8.3. Mikrokontrolery Atmel 40-pin .....	11
8.4. Mikrokontrolery Atmel AVR .....	11
8.5. Mikrokontrolery Microchip PIC.....	12
8.6. Pamięci EEPROM – SPI .....	12
<b>DODATEK A. SCHEMAT ADAPTERÓW DO PROCESORÓW AVR 89S85X5.....</b>	<b>13</b>
<b>DODATEK B. SCHEMAT ADAPTERA DLA PIC16C55, 16C57. ....</b>	<b>14</b>
<b>DODATEK C – STANDARDOWE ZŁĄCZA ISP- ATMEL.....</b>	<b>14</b>
<b>BIBLOGRAFIA.....</b>	<b>14</b>
<b>WYJAŚNIENIA.....</b>	<b>14</b>

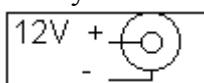
## 1. Wprowadzenie

Programator COMBO3, 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 za pośrednictwem standardowego interfejsu **RS-232**. Zasilany jest z zewnętrznego zasilacza sieciowego (stabilizowane 12V, max. 100 mA) lub z zewnętrznej baterii 12V. Cechą charakterystyczną jest obecność kilku różnych podstawek programujących, dedykowanych dla różnych układów, oraz 6-pinowego gniazda „ISP” do programowania niektórych typów układów bez wyjmowania ich z działającego urządzenia.

Niniejsza instrukcja ma za zadanie zapoznanie Użytkowników z obsługą programatora i z niektórymi możliwościami jego zastosowania.

## 2. Uruchomienie programatora

Przed włączeniem programu obsługi na komputerze, należy podłączyć programator. Do gniazda zasilającego podłączamy źródło zasilania **12V** – np. zasilacz stabilizowany 12V o wydajności min. **100mA**, zakończony standardowym rurkowym wtykiem zasilającym 2.5 / 5.5 mm :



Do połączenia z PC należy użyć standardowego kabla RS232 DCE-DTE z wtyczkami **9-pin**, łączącego wyprowadzenia 2-2, 3-3, 5-5, 7-7, 8-8 bez skrzyżowania. W przypadku gniazda **25-pin** w komputerze, wyprowadzenia DB9 programatora łączymy do DB25 komputera w inny sposób: 2-3, 3-2, 5-7, 7-4, 8-5.

Zaleca się, aby podczas uruchamiania w podstawkach programujących nie było **żadnych układów**. Niektóre rodzaje układów mogą ulec uszkodzeniu, jeśli będą w podstawce w momencie włączania zasilania.

W momencie włączania zasilania zielona dioda powinna się zapalić, natomiast czerwona krótko mignąć. Oznacza to prawidłowe funkcjonowanie urządzenia.

Po tych czynnościach należy uruchomić na komputerze właściwy program obsługi, zależnie od posiadanego sprzętu będzie to wersja programu pod Windows, DOS, Linux lub Amigę. W programie należy podać numer portu szeregowego, do którego podłączony jest programator (najczęściej **COM1** lub **COM2**). Jeżeli połączenie jest wykonane prawidłowo, program uruchamia się i wyświetla rozpoznaną wersję Firmware programatora (np. *Firmware V3.2*).

## 3. Instalowanie obsługiwanego układu w podstawce.

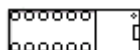
Programator COMBO3 posiada 8 podstawek programujących :

- \* 89C51 (40pin) – do programowania 40-pin układów Atmel rodziny '51, np. 89C51, 89S8252, 89C55WD
- \* AVR (20pin) – do programowania 20-pin układów Atmel 89C1051,2051,4051, 20-pin układów AVR (np. 90S1200), 8-pin AVR (np. 90S2323, ATiny), 40-pin AVR przez adapter (np. 90S8515).
- \* PIC (18pin) – do programowania układów Microchip PIC (18-pin, 8-pin, 14-pin oraz przez adapter : 28-pin, 40-pin)
- \* 24C – dla układów z serii 24Cxx, 24LCxx,
- \* 25C – dla układów z serii 25Cxx, 25LCxx,
- \* 93C – dla układów 93Cxx, 93Sxx,
- \* SDE – dla układu SDE2506
- \* ISP – do programowania w systemie – patrz rozdział 6.

**Znaki** \* na obudowie wskazują położenie **pierwszego pinu** podstawki i muszą zgadzać się z oznaczeniami na obudowie układu. **Znak \*** przy gnieździe **ISP** oznacza pierwszy pin gniazda.

**Podstawka 40-pin** służy wyłącznie dla procesorów Atmel '51 i pozostałe układy 40-pin (np. 90S8515, PIC16F877) są programowane w innych gniazdach, za pośrednictwem adapterów.

W **podstawce 20-pin** można programować 20-pin układy Atmel '51 oraz 20-pin AVR. Układy AVR (Atiny) w obudowie 8-pin należy wkładać w sposób przedstawiony na rysunku (pin 1 układu w pinie 1 podstawki):

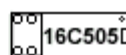
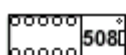


Układy Atmel AVR : **90S8515 i 90S8535** wymagają zastosowania adaptera 40 pin -> 20 pin (opis w dodatku A). Wkładany jest on do podstawki 20-pin, jak na rysunku :

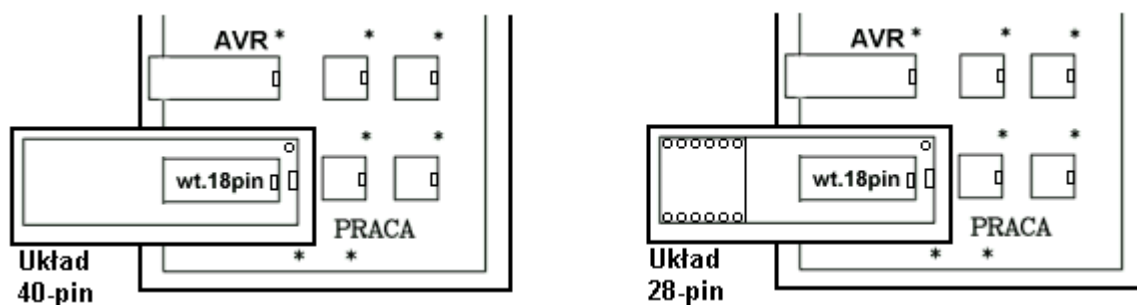


Układy Atmel AVR **ATmega** programowane są wyłącznie w trybie ISP (obudowa TQFP-64).

**Podstawka 18-pin** służy wyłącznie do układów PIC. Bezpośrednio programuje 18-pinowe układy PIC, np. 16F84, 16F628, 16C54, 16C625, układy w obudowach 8-pinowych (np. 12C508, 12C67x, 12F629) oraz 14-pinowych (16C505), lecz wkładane są one do brzegu podstawki :



Układy **40-pinowe (np. 16F877)** oraz **28-pinowe o szerokości 0.6"** (z wyjątkiem 16C55, 16C57) obsługiwane są jednym adapterem :



Układy **28-pinowe** typu **16C55, 16C57** wymagają innego rodzaju adaptera, opisanego w dodatku B.

Pozostałe 4 **podstawki 8-pinowe** dedykowane są poszczególnym rodzinom pamięci szeregowych **EEPROM** i nie stosuje się w nich żadnych adapterów.

**Gniazdo ISP (6-pin)** służy do programowania „w systemie” układów z serii PIC, AVR, 89S8252, ATmega, Atiny, szeregowych EEPROM. Za pomocą odpowiedniego kabelka łączymy programator z układem, który jest wlotowany na płytkę i którego chcemy zaprogramować. Programowanie ISP opisane jest szczegółowo w rozdziale 6.

## 4. Wybór obsługiwanego układu.

Po włożeniu układu do podstawki, należy wybrać jego typ w programie obsługi. W programie DOS wykonywane jest to za pomocą klawisza **U**, w pozostałych programach – za pośrednictwem okienek. Niektóre rodziny układów posiadają mechanizm pozwalający wykryć ich obecność **automatycznie**. Układy te to wszystkie procesory **Atmel ('51 oraz AVR)** oraz pamięci **EEPROM 24Cxx** (z wyjątkiem 24C32..24C256, które są rozpoznawane jako 24C02), Microchip PIC **16Fxxx** (tylko w programie COMBO2003). W programie DOS wystarczy wcisnąć **Alt-S**, w pozostałych : przycisk „**sygnatura**” lub „**Autodetekcja**”. Następnie program przeszukuje podstawki programujące i proponuje automatyczny wybór znalezionej podstawki.

### Tabela odpowiedników układów nie zamieszczonych w menu programu.

Niektóre układy posiadają identyczne algorytmy programowania i różnią się jedynie rozmiarami pamięci i wbudowanymi peryferiami. Z tego względu, dla zwiększenia przejrzystości programu, zrezygnowano z umieszczania w menu układów najrzadziej używanych lub wychodzących z użycia.

Układ programowany	Układ zastępczy, obecny w menu	Różnice
PIC12C508A, 12CE518	PIC12C508	kalibracja RC, EEPROM
PIC12C509A, 12CE519	PIC12C509	j.w.
PIC16C52, 16C55	PIC16C54	Rozm. EPROM
PIC16C58, 16C57	PIC16C58	Rozm. EPROM
PIC16C62, PIC16C64	PIC16C622	Rozm. EPROM, obudowa
PIC16CE625, 16C716	PIC16C622	Rozm. EPROM, obudowa
ATmega603	ATmega 103	1/2 pamięci, wycofane
ATmega64	ATmega 128	1/2 pamięci
ATmega8515, 8535	ATmega 8	Peryferia, obudowa
AT89C1051U	AT 89C1051	UART
T89C51RB2	T89C51RC2	Peryferia
AT90S4414, 4434	AT 90S85x5	1/2 pamięci, wycofane
ATmega16	ATmega 32	1/2 pamięci
ATmega323	ATmega 32	wycofany z produkcji
SDA2516	24C01	
SDA2526, PCF8582	24C02	
SDA2546, PCF8594	24C04	
SDA2586, PCF8598	24C08	
ST95xxx, M95xxx	25Cxxx	EEPROM SPI
93C14	93C46 (16-bit.)	Nazwa nieudokumentowana.
ER5911	93C46	wycofany z produkcji

## 5. Operacje na układach.

### 5.1. 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.

Jeżeli potrzebny jest inny format pliku (np. BIN, Motorola S), należy użyć konwertera – hex2bin.exe ew. bin2mot.exe

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.

## 5.2. 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).

### Uwaga – program COMBO2003 .

W pliku HEX może być dołączona informacja o ustawieniach konfiguracji układu (**CFG**), czyli **LOCK-bits** i **FUSE-bits**. Dotyczy to szczególnie układów ATmega, które posiadają wiele możliwych opcji konfiguracji. Jeżeli program napotka tą informację, odczyta ustawienia i automatycznie ustawi je po zaprogramowaniu i weryfikacji układu. Ułatwia to sprawne i seryjne programowanie układów ATmega.

## 5.3. 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. Nieprawidłowa weryfikacja oznacza błąd programowania.

## 5.4. 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  $\mu\text{W}/\text{cm}^2$  przez czas około 20 minut (dawka promieniowania 15  $\text{W}\cdot\text{s}/\text{cm}^2$  ). Kasowanie nie zmienia ustawień FUSE-bit w procesorach Atmel.

W przypadku układów **89C55WD**, **89C51RC** które były zabezpieczone, kasowanie należy przeprowadzić dwukrotnie, wyjmując układ z podstawki. Wynika to ze specyfiki tych układów. Nie jest to konieczne, jeśli układ nie był zabezpieczony.

## 5.5. Blank check

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

Jeżeli po skasowaniu układu operacja ta nie przeszła pomyślnie, oznacza to że wystąpił błąd przy kasowaniu lub układ jest uszkodzony.

## 5.6. 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 **89C55WD**, **89C51RC** zabezpieczenie jest aktywowane dopiero po wyjęciu układu z podstawki.

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 może być ustawiane automatycznie po zaprogramowaniu układu, jeśli wybrano poziom zabezpieczenia lub program (COMBO2003) odczytał informację CFG z pliku HEX.

## 5.7. Ustawianie FUSE-BIT.

Opcja dotyczy tylko układów **89S8252**, **AVR** oraz **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 FUSE-bit i zabezpieczenie kodu CP ustawiane są automatycznie po zaprogramowaniu i sprawdzeniu układu.

W programie **COMBO2003** można odczytać bieżące ustawienia układu, przyciskiem *Czytaj*.

**Należy zaznaczyć szczególną wagę tych ustawień dla procesorów PIC i AVR – mogą uniemożliwić działanie zaprogramowanego układu (np. złe ustawienie WDT, OSC, BOOT) lub nawet zablokować możliwość programowania ISP !**

**Uwaga – układy ATtiny / ATmega .**

Skasowanie bitu **!RSTDIS** (puste pole) lub ustawienie **!SPIEN** (zaznaczony) uniemożliwia programowanie w systemie (ISP) przez interfejs SPI ! Jedyne sposoby programowania jakie pozostaje, to programowanie równoległe / HV.

## 6. Programowanie ISP

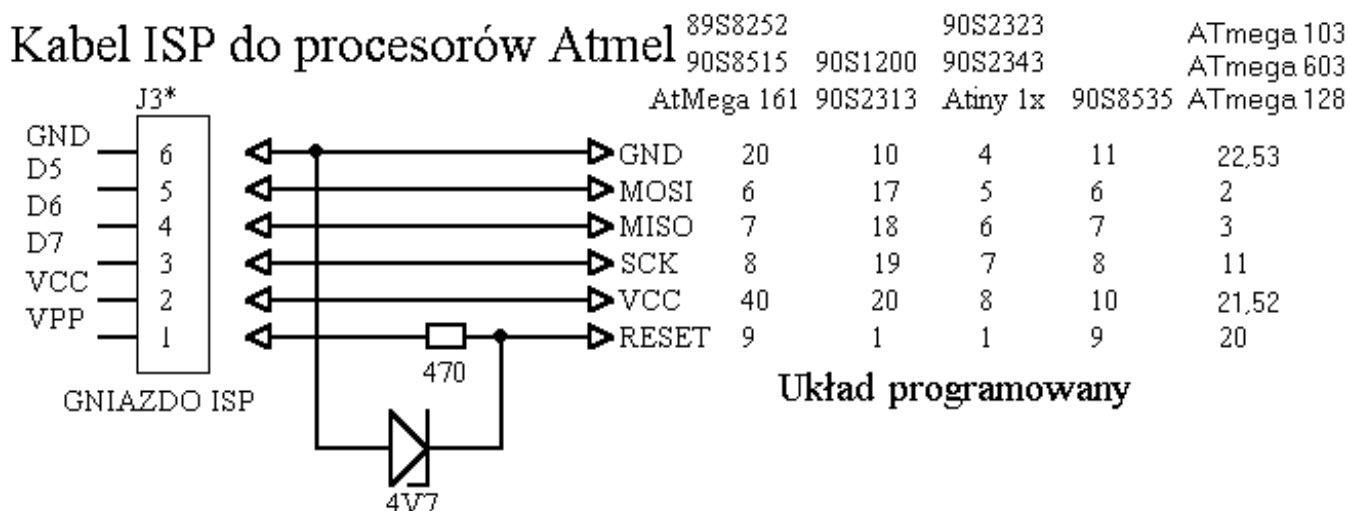
Ten sposób programowania umożliwia obsługę układów bez wylutowywania ich z urządzenia. Jest możliwy tylko dla układów **Atmel 89S8252**, **AVR**, **PIC**, **EEPROM**. Dla układów **ATmega** to jedyny, w tej chwili dostępny, sposób programowania. Zamiast wkładać układ do podstawki programatora, wystarczy połączyć go z programatorem za pomocą 6-żyłowego przewodu. Dzięki temu upraszcza się proces produkcji urządzenia, ułatwia serwis, a także unika się kosztownej podstawki SMD. Każda rodzina układów podłączana jest inaczej, dlatego wymagane jest w tej dziedzinie posiadanie odpowiedniej wiedzy. Ponieważ programowanie odbywa się szeregowo, układy Atmel są programowane wolniej w trybie ISP niż w podstawce programatora.

Układy PIC oraz EEPROM są podłączane przez ISP według poniższego opisu:

Programowanie układów "w systemie" - ISP								
PIN	Funkcja	Pin układu programowanego						
gniazda ISP		24Cxx	93Cxx	PIC16F84	PIC16F873/6	25Cxx *)	SDE 2506	PIC16F874/7
1	VPP			4	1	*)		1
2	VCC	8	8(,6-dla 16bit)	14	20	3,7,8	3	11,32
3	CLK	6	2	12	27	6	5	39
4	DAT	5	3,4	13	28	2	4	40
5	CS		1			5	2	
6	GND	4	5(,6-dla 8bit)	5	8,19	4	1,7	12,31

\*) Do 1.pinu 25Cxx należy podłączyć linię D4 z programatora (np. z 1.pinu podstawki 25Cxx lub 35.pinu podst. 40-pin)

Układy Atmel AVR oraz AT89S8252 są podłączane według schematu:



**Dioda Zenera 4,7V** i rezystor 470Ohm dla układów Atmela ma za zadanie zabezpieczyć wyprowadzenie RST układu przed napięciem +12V, które może pojawić na linii VPP np. podczas **autoidentyfikacji**. Ewentualnie można użyć też diody 5,1V lub 4,3V. Jest to konieczne **tylko dla AT89S8252** – układy PIC oraz wszystkie AVR dopuszczają +12V na RESET. Układy **89S8252** nie mogą być automatycznie rozpoznawane przy programowaniu SPI – należy je wybrać ręcznie w programie.

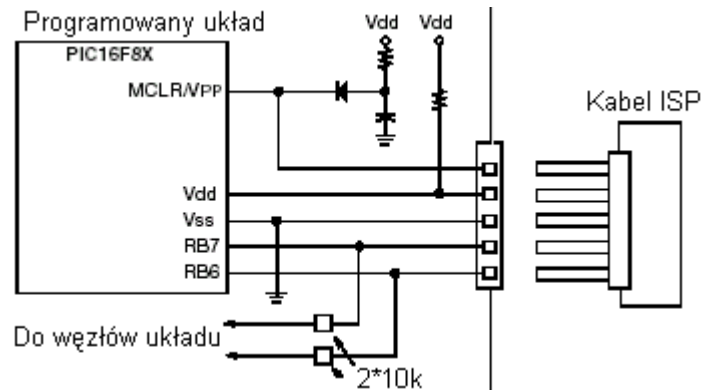
Przewód ISP nie powinien być dłuższy niż 30cm. W przypadku dłuższego przewodu, musi być zastosowany kabel dobrej jakości (ekranowany), i do prawidłowej pracy może być konieczne połączenie linii CLK i GND kondensatorem 100pF tuż przy samym układzie. Opisywany problem wystąpił z układami PIC16F877.

W niektórych konfiguracjach z AVR (np. **ATmega 128**) może okazać się konieczne połączenie RESET z GND przez rezystor 4k7, aby umożliwić wymuszenie stanu niskiego przez programator, na czas programowania.

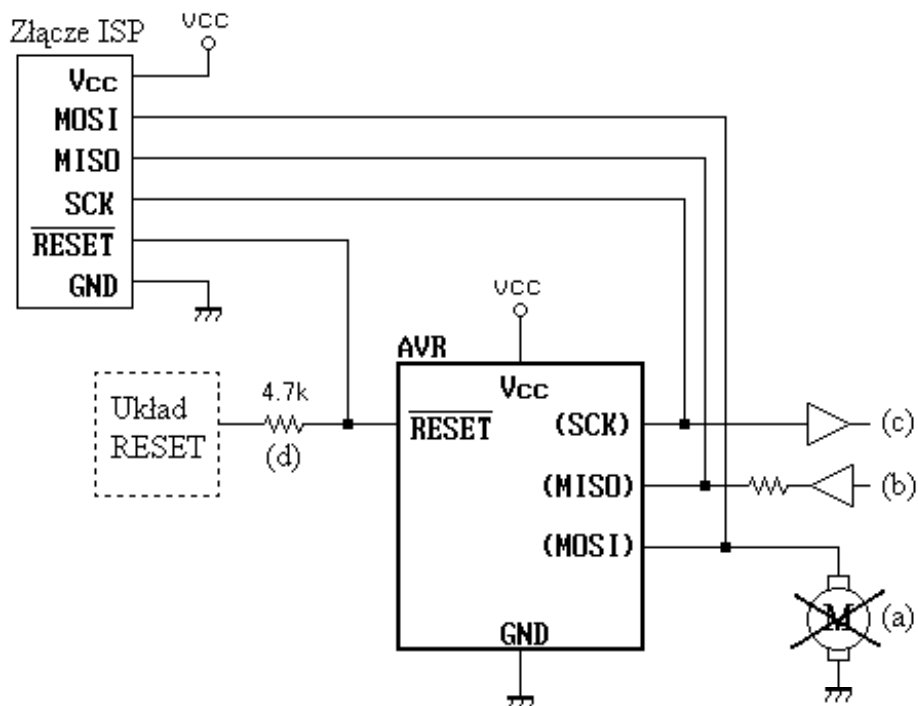
Aby programowanie ISP było możliwe, muszą być spełnione następujące warunki :

1. Układ musi umożliwiać ISP.
2. Programowanie ISP musi być **włączone** w układzie (FUSE !SPIEN=0, !RSTDIS=1 w AVR, !SPIEN=0 w 89S8252) – zazwyczaj układy dostarczane przez producenta są gotowe do ISP, lecz w niektórych aplikacjach celowo wyłącza się ten tryb w celu zwiększenia bezpieczeństwa układu i zapisanego programu.
3. Linie, do którego podłączono SPI (np. CLK, DAT, RES, MISO, MOSI ) **nie mogą być zbyt obciążone** w układzie, tak aby programator mógł swobodnie sterować ich poziomami. Nie mogą tu być bezpośrednio podłączone zworki, wyłączniki, duże pojemności ani wyjścia logiczne innych aktywnych układów na płytce. Najbezpieczniej będzie zostawić je niepodłączone, zarezerwowane dla ISP.

Przykład rozwiązania dla PIC16F84 :



Przykład rozwiązania dla Atmel AVR :



- (a) nie należy podłączać silnych obciążeń, które bezpośrednio wpływają na ISP,
- (b) nie należy sterować z zewnątrz liniami ISP, lub należy zastosować rezystor (10k),
- (c) należy się upewnić, czy sygnały ISP podczas programowania nie wpływają na inne funkcje,
- (d) rezystor uniemożliwia kolizję ISP z układem RESET.

4. **Pojemność przy lini RESET** nie powinna być większa niż 4.7 uF, **rezystancja przy lini RESET** nie powinna być mniejsza niż 4,7 kOhm. Większość układów Atmel posiada wbudowany rezystor

RESET i można go nie montować na płytce. Układy AVR, PIC mogą również pracować bez zewn. układu RESET (tylko rezystor do VCC), ponieważ posiadają wewnętrzny *Power-on RESET*. Przy układach **ATmega**, może być nawet konieczne dołączenie rezystora **4.7k między RESET a GND** w kablu ISP, ponieważ wbudowany rezystor w ATmega ma za małą wartość.

5. Kabel ISP nie może być zbyt długi. **Typowa długość to 30 cm**, czasem wymagany jest kondensator 100pF na CLK – patrz wyżej.
6. 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**.
7. Do układu musi być doprowadzone zasilanie. Najczęściej wyjście VCC z programatora jest za słabe, aby zasilić układ programowany w systemie wraz z jego otoczeniem.
8. Układy **Atmel** wymagają obecności kwarcu 1-12 MHz lub innego źródła taktowania przy ISP tak jak przy normalnej pracy układu (np. wewnętrzny generator RC). **Przy nieprawidłowym wyborze źródła zegara (np. zewnętrzny RC lub EXTCLK), należy dostarczyć przebieg taktujący 1-6 MHz z zewnętrznego generatora.**
9. W przypadku utraty synchronizacji i błędu programowania, należy wyłączyć programator, program, włączyć ponownie i spróbować jeszcze raz zaprogramować.
10. **Aby po zaprogramowaniu układ wystartował z zapisanym programem, może być konieczne rozłączenie kabla ISP. Dotyczy to jedynie szczególnych przypadków** (np. problemy z RESET lub linie SCK, MISO, MOSI, PB6, PB7 wykorzystywane w systemie docelowym).

## 8. Przegląd obsługiwanych układów

### 8.1. 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.

Spotyka się układy 93Cxx w obudowach 14-pin (nie udokumentowane przez żadnego producenta). W takim przypadku, rozkład pinów podaje tabela:

Nr pinu w obudowie 14-pin	Sygnal	Nr pinu w obudowie 8-pin
2	CS	1
3	CLK	2
5	DI	3
6	DO	4
9	GND	5
10	ORG	6
12	NC	7
13	VCC	8

Układy **93Sxx** dają odmianą 93Cxx z możliwością ustawienia różnego poziomu **zabezpieczenia przed zapisem (WP)**. Najwyższy poziom zabezpieczenia jest nieusuwalny i tak zabezpieczona kość nie może być już modyfikowana - może być jedynie czytana.

### 8.2. Mikrokontrolery Atmel 20-pin

Są to 20-nóżkowe wersje procesorów 80C51, z pamięciami FLASH na kod programu 1kB (89C1051), 2kB (89C2051), 4kB (89C4051). Możliwe do programowania tylko w podstawce programatora. Umożliwiają zabezpieczenie kodu programu przed odczytem i zapisem.

### 8.3. Mikrokontrolery Atmel 40-pin

Są to odmiany procesora 80C51 z pamięcią FLASH kodu programu : od 4kB (89C51) do 32kB (89C51RC). Umożliwiają zabezpieczenie programu przed odczytem i zapisem. Większość programowana tylko w podstawce programatora.

Na uwagę zasługuje układ AT 89S8252 : 8kB FLASH kodu programu + 2kB EEPROM na dane. Układ ten może być programowany również w trybie ISP. Programowanie ISP może być jednak zablokowane odpowiednim FUSE-bitem.

### 8.4. 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.

Szczególną podgrupę AVR stanowią układy **ATmega**. Są one programowane jedynie w trybie ISP. Są dostarczane przez producenta z włączonym wewnętrznym generatorem RC 1MHz, mają wiele możliwych konfiguracji FUSE-bit. Szczególnie dla nich w COMBO2003 powstała **opcja zapis/odczyt CFG**, **odczyt konfiguracji układu z pliku HEX**, **automatyczne ustawienie FUSE po zaprogramowaniu**.

Kolejna podgrupa AVR to układy **ATtiny**. Wszystkie mogą być programowane w trybie ISP, a niektóre - np. ATtiny12 - również w trybie HV. Są dostarczane przez producenta z włączonym wewnętrznym generatorem RC 1MHz.

### **Uwaga!**

**Wartości wszystkich bitów FUSE u Atmela są zanegowane** : 0 (pusty kwadracik) oznacza „włączony”, 1 (zaznaczony) oznacza “wyłączony”!

## **8.5. 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 HEX (standart Microchip), razem z kodem programu. Większość układów PIC może być programowana przez ISP.

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

- 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 i EEPROM (12F629, 12F675),
- 14-pinowy 16C505 – z pamięcią EPROM, wiele możliwych ustawień FUSE
- 18-pinowe 16C54, 16C68 – z pamięcią EPROM (niskobudżetowe)
- 28-pinowe 16C55, 16C57 – z pamięcią EPROM (niskobudżetowe)
- 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 żądkich 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.

## **8.6. 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, 95xxx oraz - przez podobieństwo algorytmu programowania – SDE 2506.

## Dodatek A. Schemat adapterów do procesorów AVR 89S85x5.

Numer pinu we wtyczce 20-pin	Funkcja	Nazwa pinu	Numer pinu w podstawce 90S8515	Numer pinu w podstawce 90S8535
1	RST/VPP	RST	9	9
3	RDY	PD1	11	15
5	XTAL1	XTAL1	19	13
6	IOE	PD2	12	16
7	IWR	PD3	13	17
8	BS	PD4	14	18
9	XA0	PD5	15	19
10	GND	GND	20	11
11	XA1	PD6	16	20
12	D0	PB0	1	1
13	D1	PB1	2	2
14	D2	PB2	3	3
15	D3	PB3	4	4
16	D4	PB4	5	5
17	D5	PB5	6	6
18	D6	PB6	7	7
19	D7	PB7	8	8
20	VCC	VCC	40	10

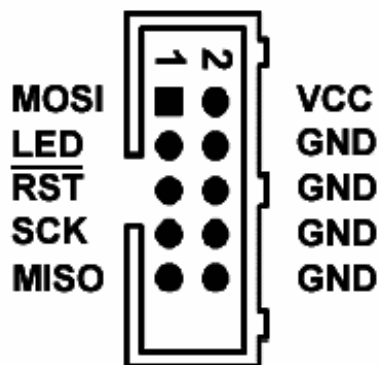
Należy wykonać połączenia wtyczki 20-pin do podstawki 40-pin (8515 albo 8535). Jako wtyczki 20-pin z powodzeniem można użyć podstawki precyzyjnej.

## Dodatek B. Adapter dla PIC16C55, 16C57.

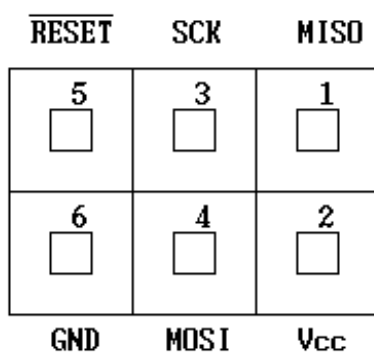
Nazwa pinu	Pin podstawki DIP-18 programator	Pin podstawki DIP-28 układu
T0CKI	3	1
RA0	17	6
RA1	18	7
RA2	1	8
RA3	2	9
RB0	6	10
RB1	7	11
RB2	8	12
RB3	9	13
RB4	10	14
RB5	11	15
RB6	12	16
RB7	13	17
OSC1	16	27
MCLR	4	28
VDD	14	2
VSS	5	4

## Dodatek C. Standardowe złącza ISP- Atmel.

Znajdują one zastosowanie przy układach AVR / ATmega / ATtiny (standard STK-200, STK-500).



Gniazdo IDC-10 pin.



Gniazdo IDC-6 pin.

- LED – sygnał aktywny podczas programowania (sterowanie diodą LED i podłączenie linii do układu w STK-500).
- Oba typy gniazd mają raster 2.54 mm.

## Bibliografia

1. Strona domowa programatora : <http://www.progstar.com.pl/combo3>
2. Oprogramowanie pod Windows : <http://www.combosoft.republika.pl>
3. Oprogramowanie dla Amigi : <http://www.comboamiga.republika.pl>
4. Strony katalogowe i noty aplikacyjne firmy Atmel: <http://www.eu.atmel.com>
5. Strony katalogowe i noty aplikacyjne firmy Microchip: <http://www.microchip.com>
6. A.Rydzewski : *Mikrokontrolery jednoukładowe rodziny MCS-51*

## Wyjaśnienia.

Producent i sprzedawca COMBO3 nie ponoszą żadnej odpowiedzialności za szkody wynikłe z niewłaściwej eksploatacji bądź awarii tego urządzenia.

Sprzedawca zestawów do samodzielnego zmontowania COMBO3 nie ponosi odpowiedzialności za nieprawidłowy montaż zestawu przez Nabywcę i ewentualne straty z tym związane.

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

Producent i sprzedawca COMBO3 nie odpowiada za żadne szkody, które powstały przez nielegalne zastosowanie programatora COMBO3.

Producent i sprzedawca COMBO3 nie odpowiada za ewentualną utratę danych powstałą na skutek awarii programatora.

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